



**POLITECHNIKA
GDAŃSKA**

WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI I INFORMATYKI



Imię i nazwisko autora rozprawy: Krzysztof Czuszyński
Dyscyplina naukowa: Biocybernetyka i inżynieria biomedyczna

ROZPRAWA DOKTORSKA

Tytuł rozprawy w języku polskim: Rozpoznawanie statycznych i dynamicznych gestów dłoni z zastosowaniem adaptacyjnego czujnika optycznego

Tytuł rozprawy w języku angielskim: Recognition of static and dynamic hand gestures utilizing an adaptive optical sensor

Promotor

podpis

dr hab. inż. Jacek Rumiński, prof. nadzw. PG

Gdańsk, rok 2018

Streszczenie rozprawy w języku polskim:

Bezdotykowe interfejsy urządzeń stacjonarnych oraz urządzeń mobilnych są ważne zwłaszcza w ochronie zdrowia (np. przeciwdziałanie roznoszeniu bakterii i wirusów, wsparcie chirurga w trakcie przeprowadzania operacji itp.). W ramach rozprawy doktorskiej zaprojektowano i wykonano oszczędny energetycznie czujnik gestów o budowie liniowej, złożony z fotodiod i diod LED, wraz z dedykowanym mu zestawem metod rozpoznawania gestów dłoni. Podczas badań pozytywnie zweryfikowano możliwość wykorzystania dwóch trybów pracy czujnika, pasywnego i aktywnego, różniących się załączeniem podświetlenia dłoni przez diody LED. Określono warunki pracy odpowiednie dla trybu pasywnego pozwalającego na oszczędność ok. 93% mocy w stosunku do trybu aktywnego. Zaproponowano 2 metody detekcji statycznych póz dłoni. Metoda bazująca na sztucznych sieciach neuronowych pozwoliła na osiągnięcie dokładności klasyfikacji trzech póz dłoni na poziomie 98,76% w trybie pasywnym i 93,46% w trybie aktywnym. Do rozpoznawania gestów jako sekwencji póz wykorzystany został model rekurencyjnych sieci neuronowych. Wykazano, iż spośród 3 sposobów przygotowania danych, model wytrenowany na danych nieprzetworzonych odznacza się najwyższą dokładnością klasyfikacji 27 gestów dla częstotliwości próbkowania czujnika co najmniej 50Hz. Uzyskana wysoka dokładność klasyfikacji gestów dla częstotliwości próbkowania 100Hz (96,89%) i nieduże zużycie energii (4,96 mA) pozwalają na rozważenie powstałego czujnika jako interfejsu dla różnych rozwiązań konstrukcyjnych (np. okularów multimedialnych lub kiosków informacyjnych).

Streszczenie rozprawy w języku angielskim:

Contactless interfaces of stationary devices and mobile devices are important especially in health care (eg preventing the spread of bacteria and viruses, surgeon support during surgery, etc.). As part of the doctoral thesis, an energy-effective gesture sensor with a linear structure was designed and made, consisting of photodiodes and LED diodes, along with a set of hand gesture recognition methods. During the tests, the possibility of using two operating modes, passive and active, differing with the utilization of the LEDs for hand backlight, was positively verified. The working conditions for the passive mode, which saves about 93% of power in relation to the active mode, have been determined. Two methods for the detection of static hand poses have been proposed. The method based on artificial neural networks allowed to achieve the accuracy of classification of three hand poses at the level of 98.76% in passive mode and 93.46% in active mode. To recognize gestures as a sequence of poses, a recurrent neural networks model was used. It was shown that among three methods of data preparation, the model trained on raw data has the highest classification accuracy of 27 gestures for the sampling frequency of at least 50 Hz. The obtained high accuracy of gestures classification for sampling frequency 100 Hz (96.89%) and low energy consumption (4.96 mA) allow to consider the resulting sensor as an interface for various construction solutions (eg smart glasses or interactive kiosks).

Podziękowania

*Dziękuję Promotorowi, **dr hab. inż. Jackowi Rumińskiemu, prof. nadzw. PG** za prowadzenie w badaniach, motywowanie do dobrego tempa pracy, wnikliwą analizę i bezkompromisowe oczekiwania.*

*Dziękuję **Profesorowi Jerzemu Wtorkowi** za zachętę do podjęcia i otrzymanie szansy rozwoju w ramach studium doktoranckiego oraz za wieloletnią współpracę.*

*Dziękuję **dr Adamowi Bujnowskiemu** za nieocenioną pomoc w projektowaniu i uruchamianiu urządzeń elektronicznych.*

*Dziękuję szczególnie swoim **Rodzicom** a także całej **rodzinie** oraz **przyjaciołom**, w tym **koleżankom i kolegom z pracy**, za każdą pomoc, każde słowo zachęty i wsparcie, które otrzymałem w okresie swojej pracy nad niniejszą rozprawą doktorską.*

Spis treści

1. Wprowadzenie	9
1.1. Bezdotykowe interfejsy gestów w urządzeniach mobilnych, w przemyśle i w innych zastosowaniach.....	14
1.2. Rola bezdotykowych interfejsów gestów w ochronie zdrowia	15
1.3. Taksonomie gestów.....	16
1.4. Wymagania użytkowe i techniczne interfejsu.....	21
1.5. Cel, tezy i zakres pracy	23
1.6. Układ pracy	25
2. Czujniki i metody rozpoznawania gestów	27
2.1. Typy czujników gestów	28
2.1.1. Optyczne czujniki gestów.....	28
2.1.2. Czujniki gestów inne niż optyczne	42
2.1.3. Podsumowanie.....	44
2.2. Rozpoznawanie gestu jako proces.....	44
2.3. Metody rozpoznawania póż i gestów	46
2.3.1. Metody klasyfikacji jednej obserwacji	48
2.3.2. Metody klasyfikacji sekwencji	53
2.4. Miary efektywności metod klasyfikacji	57
2.5. Podstawowy zbiór póż dłoni	58
2.6. Projekt zbioru akcji do obsługi urządzeń mobilnych	59
2.7. Podsumowanie rozdziału.....	62
3. Optyczny czujnik gestów o liniowej budowie	65
3.1. Koncepcja i zasada działania czujnika gestów o liniowej budowie	66
3.2. Realizacje liniowego sensora gestów	68
3.3. Parametryzacja i wymagania dla liniowego sensora gestów.....	71
3.3.1. Grupa kontrolna.....	72
3.3.2. Spłaszczenie profilu podświetlenia	72
3.3.3. Strefy sensora	73
3.3.4. Zdolność rozdzielcza	75
3.3.5. Czułość przestrzenna i czasowa	76
3.3.6. Pobór prądu	77
3.4. Metodologia ewaluacji właściwości optycznego sensora gestów	78

3.4.1.	Stanowisko laboratoryjne	78
3.4.2.	Badania symulacyjne działania sensora optycznego	81
3.4.3.	Weryfikacja wyników symulacji względem pomiarów	88
3.4.4.	Zgodność symulacji i pomiarów.....	90
3.5.	Wyniki pomiarów i symulacji.....	92
3.5.1.	Grupa kontrolna.....	92
3.5.2.	Splaszczanie profilu podświetlenia	94
3.5.3.	Strefy sensora	95
3.5.4.	Zdolność rozdzielcza	95
3.5.5.	Czułość przestrzenna i czasowa	97
3.5.6.	Pobór prądu i kalibracja.....	99
3.6.	Możliwości zastosowania czujnika do obsługi gestów ciągłych.....	103
3.6.1.	Nieliniowości odwzorowania położenia dłoni.....	103
3.6.2.	Utrzymanie dynamiki zmian amplitudy sygnału.....	104
3.7.	Podsumowanie rozdziału.....	106
4.	Klasyfikacja póż i gestów z użyciem optycznego czujnika o liniowej budowie	111
4.1.	Implementacja GRS	111
4.2.	Klasyfikacji póż.....	115
4.2.1.	Baza póż dłoni	115
4.2.2.	Metoda I: dopasowanie krzywych wzorca i obserwacji	117
4.2.3.	Metoda II: parametryzacja i sztuczne sieci neuronowe	124
4.2.4.	Pozy rozpostarte	128
4.2.5.	Porównanie metod klasyfikacji póż zwartych	128
4.3.	Zachowanie sensora w różnych warunkach oświetlenia	132
4.3.1.	Walidacja czujnika liniowego w trybie pasywnym	133
4.3.2.	Baza póż dłoni dla dwóch trybów pracy i klasyfikacja póż	135
4.3.3.	Układ pomiarowy z platformą obrotową.....	137
4.3.4.	Charakterystyka czujnika liniowego względem warunków oświetlenia	140
4.3.5.	Analiza możliwości automatycznego przełączania trybu pracy czujnika optycznego	146
4.3.6.	Dyskusja i podsumowanie	148
4.4.	Rozpoznawanie gestów z wykorzystaniem rekurencyjnych sieci neuronowych	153
4.4.1.	Baza gestów dłoni.....	153
4.4.2.	Uczenie modelu RNN.....	159
4.4.3.	Wyniki ewaluacji metod rozpoznawania gestów	162

4.5. Podsumowanie rozdziału.....	169
5. Praktyczne zastosowania czujnika optycznego o liniowej budowie i wnioski końcowe.....	172
5.1. Podsumowanie opracowanych rozwiązań.....	172
5.2. Opracowane przykłady zastosowań	175
5.2.1. Czujnik optyczny jako dołączany moduł.....	175
5.2.2. Czujnik optyczny wbudowany w okulary multimedialne	177
5.3. Przyszłe prace.....	180
5.3.1. Badania nad praktycznym użytkowaniem czujnika gestów	180
5.3.2. Możliwości dalszego zastosowania i rozwoju	182
5.3.3. Praktyczne możliwe zastosowania w ochronie zdrowia.....	183
5.4. Podsumowanie rozprawy doktorskiej	186
5.5. Oryginalne osiągnięcia.....	188
Bibliografia	191
Dodatek A: Dorobek naukowy mgr. inż. Krzysztofa Czuszyńskiego.....	201
Dodatek B Schemat PCB	205

WYKAZ WAŻNIEJSZYCH OZNACZEŃ

Symbol	Rozwinięcie	Znaczenie
d_{LED}	LEDs distance	Odległość pomiędzy sąsiednimi LEDami czujnika
d_{PD}	Photodiodes distance	Odległość pomiędzy sąsiednimi fotodiodami czujnika
f_s	Sampling frequency	Częstotliwość próbkowania
h	Height	Odległość od czujnika wzdłuż osi y
h_{bLED}	Height of optical block of a LED	Wysokość bloku optycznego diody LED
h_{bPD}	Height of optical block of a photodiode	Wysokość bloku optycznego fotodiody
I_D	Indentation depth	Głębokość wcięcia w profilu intensywności światła widzianego przez sensor
N_f		Liczba fotonów biorąca udział w symulacji Monte Carlo
n_{LAY}	Number of layers	Ilość warstw ukrytych sieci neuronowej
n_{LED}	Number of LEDs	Liczba diod LED obecnych w czujniku
n_{NEU}	Number of neurons	Ilość neuronów w warstwie ukrytej sieci neuronowej
n_{PD}	Number of photodiodes	Liczba fotodiod obecnych w czujniku
R	Radius	Promień obiektu interakcji (palca) w modelu Monte Carlo
$shift$		Odległość środka czołowej strony obiektu od środka czujnika wzdłuż osi x
V_{not}	Noticeable velocity	Maksymalna wykrywalna prędkość przemieszczenia obiektu przez sensor operujący z częstotliwością f_s
w	Width	Szerokość obiektu interakcji (np. palca lub układu złączonych palców) w modelu Monte Carlo
β_{LED}		Pole widzenia diody LED
β_{PD}		Pole widzenia fotodiody

WYKAZ WAŻNIEJSZYCH SKRÓTÓW I POJĘĆ

Skrót	Rozwinięcie	Znaczenie
ADC	Analog to Digital Converter	Konwerter analogowo cyfrowy
ANN	Artificial Neural Network	Sztuczna sieć neuronowa
FJ	Fingers Joined	Palce złączone
FOV	Field Of View	Pole widzenia
FS	Fingers Separated	Palce rozłączone
FSM	Finite State Machine	Skończona maszyna stanów
GRS	Gesture Recognition System	System rozpoznawania gestów
GUI	Graphical User Interface	Graficzny interfejs użytkownika
HCI	Human Computer Interaction	Interakcja człowieka z komputerem
HMM	Hidden Markov Model	Ukryty model Markowa
MCMS	Monte Carlo Method Simulation	Symulacja Monte Carlo
MGS	Motion Gesture Sensor	Czujnik gestów ruchowych
NTE	Near To Eye (Display)	Bliski oku (wyświetlacz)
PCB	Printed Circuit Board	Płytko drukowana
RNN	Recurrent Neural Network	Rekurencyjna sieć neuronowa
RSS	Residual Sum of Squares	Resztkowa suma kwadratów
UART	Universal Asynchronous Receiver Transmitter	Uniwersalny asynchroniczny nadajnik-odbiornik

1. WPROWADZENIE

Rynek osobistych, powszechnie dostępnych przenośnych urządzeń komunikacyjnych zaczął kształtować się w połowie XX wieku w Stanach Zjednoczonych. Pionierskim rozwiązaniem mobilnym, jeśli chodzi o adaptację do użytku masowego, wprowadzonym w latach 50, były pagery. Stały się one pierwszym przełomem komunikacyjnym, pozwalając na bezprzewodową wymianę informacji w formie krótkich wiadomości tekstowych między ludźmi na odległość. Pagery upowszechniły się między innymi dzięki niewielkim rozmiarom oraz niedużemu zużyciu energii pozwalającemu na działanie przez kilka tygodni. Szacuje się, że w szczytowym okresie ich popularności, w roku 1994, na świecie w użytku było ok. 61 milionów pagerów¹.

Jednak zwłaszcza na przestrzeni ostatnich dwóch dekad rynek urządzeń mobilnych ulegał szczególnie istotnym transformacjom wynikającym z przełomów technologicznych i koncepcyjnych. Obniżenie początkowo wysokiej ceny zarówno abonamentów jak i samych komórkowych aparatów telefonicznych oraz minimalizacja sprzętowa i ograniczenie znacznej wagi tych urządzeń pozwoliły na upowszechnienie wynalazku telefonu przenośnego. Kiedy około roku 1993 waga telefonów komórkowych zaczęła spadać poniżej 250g oraz znacznie zmniejszyły się ich rozmiary, stały się one prawdziwie osobistymi urządzeniami przenośnymi. Daleko bardziej zaawansowane upowszechnienie się telefonów w stosunku do pagerów potwierdzają statystyki. W 2016 roku liczba aktywnych

¹ <http://inventors.about.com/od/pstartinventions/a/pager.htm>

użytkowników telefonów na świecie szacowana była na 4,61 mld osób co stanowiło ok. 62% ówczesnej populacji.

Kolejna rewolucja, tym razem z pogranicza telekomunikacji i elektroniki użytkowej, rozpoczęła się w roku 2007 kiedy firma Apple przedstawiła światu pierwsze urządzenie typu telefon multimedialny (smartfon, ang. *smartphone*) o nazwie iPhone. Dotychczasowa koncepcja standardowego telefonu o małym ekranie i układzie wielu fizycznych klawiszy ustąpiła konstrukcji składającej się głównie z dużego, kolorowego i dotykowego wyświetlacza. Funkcja obsługi dotyku przez ekran zdefiniowała w skali globalnej nowy typ interakcji z treścią prezentowaną na ekranie. Wykorzystanie wybierania opcji na wyświetlaczu przez bezpośrednie naciśnięcie danej kontrolki czy obsługa gestów wielopunktowych (ang. *multi touch*) wymagało projektowania nowego typu graficznych interfejsów użytkownika (ang. *graphical user interface*, GUI). Niedługo po prezentacji iPhone'a, w ramach odpowiedzi konkurencji, firmy Google, pod koniec roku 2008 zaprezentowany został pierwszy telefon (HTC, model Dream lub G1) z systemem operacyjnym Android, obecnie wykorzystywanym przez większość producentów smartfonów. Właśnie w tym okresie na dobre rozpoczęła się era smartfonów czyli telefonów, jako urządzeń służących nie tylko do komunikacji ale też mogących wspierać użytkownika w wielu codziennych czynnościach jak np. przez robienie zdjęć, nagrywanie filmów, dostęp do Internetu, funkcję poziomicy, funkcję nawigacji GPS i wiele innych. Wraz z rozwojem technologii kolejne generacje telefonów multimedialnych otrzymują coraz więcej sensorów o coraz lepszych parametrach (np. kamery o wyższej rozdzielczości, czytniki linii papilarnych, kamery termalne, kamery głębi i inne) zapełniając przy tym, dzięki wsparciu programistów, kolejne nisze użytkowe i umożliwiając nowe metody interakcji człowiek-komputer. Mimo wymienionego szeregu zalet, istnieją jednak czynności, w trakcie wykonywania których smartfony nie są naturalnym wsparciem dla użytkownika (np. długotrwałe skanowanie kodów kreskowych poprzez trzymanie telefonu w dłoni).

Realna szansa na wyjście naprzeciw takim niedogodnościom jak zajęte ręce użytkownika urządzenia mobilnego pojawiła się w roku 2012. Stało się jasne, iż technologia umożliwia formy interakcji uwalniające w większym stopniu dłoń użytkownika dzięki wyeksportowaniu podręcznego urządzenia mobilnego do formy elektronicznych okularów. W tym okresie pojawiły się bowiem pierwsze doniesienia na temat projektu Google Glass. Były to pierwsze tak zminiaturyzowane okulary elektroniczne z segmentu okularów

multimedialnych (ang. *smart glasses*), posiadające swój procesor, zminiaturyzowany interfejs oraz półprzezroczysty ekran typu NTE (ang. *Near To Eye*) mogący wyświetlać informacje kontekstowe. Koncept i realizacja prawdziwie funkcjonalnych i przenośnych okularów multimedialnych doczekał się wtedy odpowiedniego rozwinięcia technologii i otrzymał szansę by realnie zaistnieć w przestrzeni publicznej.

W tym samym roku 2012, przez Katedrę Inżynierii Biomedycznej z Wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej, zgłoszony został do funduszu ERA-NET-CHIST-ERA II międzynarodowy projekt *Interaktywne okulary dla mobilnego i percepcyjnego przetwarzania (eGlasses)*. W ramach przyznanego grantu finansowanego także przez przez NCBiR, FWF, SNSF, ANR, i FNR, w latach 2014-2016 w czterech ośrodkach współpracujących w projekcie *eGlasses*, prowadzone były badania nad różnymi metodami interakcji użytkownika z okularami multimedialnymi oraz z obiektami otoczenia za pomocą okularów. W trakcie prac badawczych powstała nowa, otwarta platforma elektronicznych okularów wraz z szeregiem czujników, wykorzystywanych również jako interfejsy użytkownika. Wraz z powstaniem koncepcji urządzeń typu multimedialnych okularów jak eGlasses czy Google Glass zaistniała potrzeba rozwoju badań nad nowymi metodami interakcji człowieka z urządzeniami wyposażonymi w funkcjonalne systemy operacyjne a pozbawionymi klawiatury, myszy czy ekranu dotykowego.

W podobnym czasie w stosunku do powstania pierwszych komercyjnych okularów multimedialnych rozwinęła się także istotnie branża gogli do wirtualnej rzeczywistości (ang. *Virtual Reality, VR*). Przez wiele lat do oglądania sztucznego obrazu dookólnego dedykowane były specjalne gogle połączone z wydajnym obliczeniowo komputerem PC. Jednak w ostatnich latach technologiczny rozwój pozwolił także na zaadaptowanie do tego celu smartfonów o najwydajniejszych procesorach. Przy pomocy specjalnego uchwytu z zestawem soczewek i oprogramowania telefonu wyświetlającego odświeżany z wysoką częstotliwością obraz osobno dla każdego oka wrażenie oglądania obrazu w trójwymiarze stało się możliwe bez użycia komputera i dodatkowych przewodów łączących okulary.

Zanim urządzenia takie jak przenośne gogle VR czy zwłaszcza multimedialne okulary zaistniały na rynku konsumenckim, wyzwaniem szczególnej uwagi dla projektantów tego typu urządzeń był rozwój interakcji na linii człowiek – urządzenie mobilne, obszaru będącego fragmentem szerokiej dziedziny jaką jest interakcja człowieka z komputerem (ang. *Human Computer Interaction, HCI*). Lumsden i Brewster stwierdzili,

że do użytkowania elektroniki ubieralnej potrzebna jest zmiana paradygmatu polegającego na stosowaniu technik interakcji z urządzeniami mobilnymi poprzez eksplorację nowych metod interakcji (u nich przez głos i gesty) [1]. W innym badaniu rozważali oni natomiast metody interakcji z elektroniką ubieralną z użyciem czujnika ruchów głowy oraz czujnika gestów palcem [2]. W późniejszym okresie, badania nad interakcją z telefonem za pomocą gestów wykonywanych bezpośrednio nad nim przeprowadzili Freeman i in. [3]. Serrano i in. rozważali z kolei możliwości wykorzystania gestów dłoni do obsługi wyświetlaczy noszonych na głowie (ang. *Head-Worn Displays*) [4].

W okularach multimedialnych czy goglach VR (zwłaszcza bazujących na smartfonach), które najczęściej pozbawione są klawiatury fizycznej a nawet i panelu dotykowego, jednym z kluczowych modułów funkcjonalnych jest właśnie interfejs człowiek–maszyna. Urządzenia, które wyposażone są w mikrofon, wykorzystując odpowiednie interpretatory mowy mogą być sterowane głosem. Tego typu kontrola może być wystarczająca kiedy użytkownik zechce wyzwoić konkretną komendę lecz może nie być wygodnym rozwiązaniem choćby kiedy zajdzie potrzeba określenia pewnej wartości z danego zakresu, bądź sprawnego przeniesienia zaznaczenia aktywnej kontrolki o kilka ikon w daną stronę. Dlatego interakcja z graficznym interfejsem użytkownika realizowana jest na wiele sposobów. Obecnie w tym celu w okularach multimedialnych stosuje się oprócz rozpoznawania mowy (np. Optinvent ORA-2) także interpretatory ruchów gałki ocznej (np. Tobii Pro Glasses 2), gestów rąk na podstawie obrazu z kamer (np. Microsoft HoloLens), ruchów głowy na podstawie akcelerometrów (np. Lumus DK-51) a z interfejsów dotykowych np. sprzętowych guzików (np. Vuzix M300), dotykowego pojemnościowego panelu na ramce okularów (np. Google Glass), interfejsu inteligentnej tkaniny (np. proCover, FlexTiles w eGlasses) czy dodatkowego urządzenia jak połączony przewodem z okularami touchpad (np. Epson Moverio BT-200).

Wobec rozwoju technologii i możliwości obliczeniowych procesorów dla urządzeń mobilnych, warta rozważenia stała się również idea polegająca na opracowaniu współpracującego zestawu interfejsów dla platform przenośnych. Może bowiem zdarzyć się, że użytkownik nie będzie w stanie w danym momencie obsłużyć danej metody wprowadzania komend sterujących (np. wypowiedzieć komendy głosowej, skinąć głową, wykonać ruchu dłonią lub wykonać gestu gałką oczną) [5]. Atrakcyjna idea fuzji interfejsów na platformie mobilnej wymaga jednak zadbania o efektywność obliczeniową poszczególnych sensorów

biorąc pod uwagę, że mogą one stanowić szczególnie funkcjonalne zabezpieczenie użytkownika kiedy będą uruchomione równocześnie. Dobrym rozwiązaniem wydaje się być odciążenie platformy głównej urządzenia mobilnego w sposób sprzętowy. Przykładowo układ czujnika gestów mógłby zawierać własny zintegrowany mikroprocesor, pobierający minimalną ilość mocy i odpowiedzialny za przetwarzanie danych i wysyłanie jedynie krótkich komunikatów do sterowanego urządzenia. Zalecenia sformułowane w niniejszym wprowadzeniu mogą być pierwszymi dobrymi praktykami podczas etapu projektowania interfejsów do urządzeń mobilnych.

Okulary multimedialne, mimo, że mogą posiadać wiele z zalet smartfonów (np. system operacyjny otwarty na własne aplikacje, sensory różnego typu) oraz szereg własnych przewag (np. brak konieczności angażowania rąk jedynie do trzymania urządzenia, wyświetlacz zawsze w polu widzenia), mogą nie upowszechnić się do takiego stopnia jak smartfony. Główną przyczyną wydaje się być kwestia ochrony prywatności, gdyż kamery wbudowane w okulary multimedialne mogą być wykorzystywane do nagrywania ludzi znajdujących się w ich polu widzenia, a przede wszystkim rozmówców, bez ich wiedzy czy zgody. Wraz z wprowadzeniem w ostatnich latach rozwiązań takich jak przede wszystkim Google Glass, generacji Epson Moverio BT czy Vuzix M tego typu zagadnienia stały się aktualne. Jednak z drugiej strony, cechy użytkowania okularów multimedialnych czynią je atrakcyjnym rozwiązaniem mogącym wspierać wybrane czynności zawodowe. Jedną z zalet użytkowych takich okularów np. w przemyśle są choćby wolne ręce użytkownika. Praktycznym przykładem wykorzystania tej cechy może być praca w ręcznej sortowni paczek do wysyłania. Skanowanie kodów kreskowych podczas ręcznej weryfikacji paczek, niezależnie czy dokonywane smartfonem czy dedykowanym czytnikiem, wymaga bowiem użycia dłoni (do trzymania urządzenia) co może powodować konflikt z inną czynnością - przemieszczania oburącz większej paczki. Okulary multimedialne z wbudowaną kamerą pozwalają zaś na zeskanowanie kodu znajdującego się w odpowiednim obszarze ich pola widzenia jedynie poprzez odpowiednie zwrócenie głowy z okularami i kamerą w tym kierunku, pozostawiając obydwie dłonie gotowe do pracy.

1.1. Bezdotykowe interfejsy gestów w urządzeniach mobilnych, w przemyśle i w innych zastosowaniach

Istnieje szereg potrzeb na bezkontaktową interakcję z urządzeniami elektronicznymi w życiu codziennym i przemyśle. Mogą to być praca w kuchni, w warsztacie, praca w ogrodzie lub polu przy systemach wspomagających uprawę roślin, w laboratorium chemicznym, na halach produkcyjnych, w serwisach naprawczych, w manufakturach, tam gdzie kontakt dłoni z urządzeniem jest niewskazany ze względu na wyładowania elektrostatyczne, bezpieczne włączanie światła w łazience itp. Potrzeba cechy bezkontaktowości interfejsu z urządzeniem uwypukla się też w sytuacjach często spotykanych w przemyśle, kiedy użytkownik posiada dłonie zabrudzone w takim stopniu, że obsługa typowego dotykowego urządzenia elektronicznego nie powinna być brana pod uwagę. Mogą to być scenariusze pracy na platformach wiertniczych, na placu budowy czy podczas naprawy skomplikowanych układów i maszyn przez serwisantów. Przykładowo, ci ostatni, wyposażeni w multimedialne okulary za pomocą gestów mogliby aktywować kolejne instrukcje lub przełączać obrazy objaśniające rozwiązanie danego problemu. W podobny sposób określane są przez producenta zalety komercyjnych okularów Vuzix M100. Do zastosowań przemysłowych zalecana była także prototypowa wersja okularów obsługująca rzeczywistość rozszerzoną Vuzix M2000AR. W kontekście stosowania gogli VR użytkownik może zaś przypuszczalnie osiągnąć poczucie głębszego zanurzenia w świecie wirtualnym przy interakcji z wyświetlaną zawartością za pomocą bezdotykowych gestów niż używając fizycznych interfejsów.

Rozważając interfejsy bezdotykowe poza kontekstem wykorzystania w urządzeniach przenośnych innym scenariuszem ich zastosowania może być kontrola treści wideo na ekranie za pomocą gestów w kuchni. Zestaw prostych gestów mógłby pozwolić kucharzowi podczas przygotowywania posiłku na odpowiednie sterowanie filmem z przepisem na daną potrawę.

Bezdotykowa obsługa systemu operacyjnego komputera pokładowego jest też coraz bardziej powszechnie stosowana przez producentów samochodów. Dziedzina ta nosi nazwę interakcji człowieka z pojazdem (ang. *Human Vehicle Interaction*, HVI). W 2014 roku na targach CES zaprezentowane zostały koncepcyjny kokpit Visteon Horizon z możliwością obsługi części funkcjonalności systemu operacyjnego samochodu za pomocą gestów. Również wersje samochodów dostępnych na rynku konsumenckim, takie jak Cadillac XTS czy Chevrolet Corvette wyposażone zostały w interfejsy gestów [6]. Obsługę bezdotykową

oferuje także specjalna wersja VW Golf R Touch [7]. Zastosowanie tego typu metod interakcji z pojazdami, nie tylko dla samochodów osobowych lecz i ciężarowych, może zwiększać wygodę użytkownika lecz także i bezpieczeństwo prowadzenia pojazdów [8].

1.2. Rola bezdotykowych interfejsów gestów w ochronie zdrowia

Interfejsy bezdotykowe odgrywają szczególną rolę w zastosowaniach medycznych. Przykładowo, używane jako metoda interakcji z okularami multimedialnymi, mogą zostać wykorzystane choćby na sali operacyjnej, jako wsparcie chirurga. Okulary z wyświetlaczem zamieszczonym blisko oka mogą bowiem szybko i bezpośrednio dostarczać lekarzowi precyzyjnych informacji związanych z zabiegiem, np. dotyczących tętna pacjenta czy zdjęć chorych fragmentów operowanego organu. Osobiste przełączanie przez chirurga podglądu kolejnych opcji czy wyświetlanych zdjęć w sytuacjach wymagających zachowania sterylności dłoni powinno być zaś realizowane jedynie poprzez interfejsy bezkontaktowe (np. gest lub mowę) [5][9].

Bezdotykowe interfejsy na sali operacyjnej w trakcie zabiegu były także stosowane przez chirurgów oglądających treści związane ze stanem zdrowia pacjenta na dużym ekranie stojącym w pobliżu stołu operacyjnego. W takim przypadku często wykorzystywanym interfejsem gestów, także zapewniającym zachowanie warunków septycznych, były kamera Microsoft Kinect [10][11][12][13] lub kamery głębi innego typu [14] np. Leap Motion [15].

Możliwość stosowania okularów multimedialnych w trakcie operacji została przykładowo wykorzystana do przeprowadzenia szkoleń na żywo z zabiegu, dla osób znajdujących się w innej sali, dzięki transmisji głosu i obrazu widzianego przez lekarza z jego perspektywy [16]. W tym zastosowaniu za pomocą gestów możliwe byłoby np. powiększanie odpowiednich fragmentów transmitowanego obrazu bądź zaznaczanie ich celem uwypuklenia informacji istotnej dla widzowi.

System okularów multimedialnych z bezdotykowym interfejsem mógłby być także stosowany przez cały zespół lekarzy, anestezjologów i pielęgniarek w trakcie szczególnie złożonych i długich zabiegów, gdzie kontrola parametrów i wskaźników życiowych pacjenta mogła by być dzięki temu wielokrotnie zabezpieczona.

Analizę literaturową zastosowań bezdotykowych interakcji z oprogramowaniem radiologicznym i w trakcie operacji wykonał Mewes i in. [17]. Wskazują oni na to, że w okresie do 2016 roku łącznie, z 55 znalezionych publikacji związanych z tą dziedziną 62% z nich dotyczyło kontroli obrazów medycznych w przeglądarkach, 13% opisywało

asystę laparoskopową zaś po 9% dotyczyło asysty telerobotycznej i sterowania na sali operacyjnej. Jednak tylko 8 (tj. 14,5%) z opisanych systemów zostało przetestowanych w rzeczywistych warunkach klinicznych.

Ważnym zastosowaniem interfejsów bezdotykowych powinny być też miejsca użytku publicznego. Poza szpitalami, w których w jako pierwszych można spodziewać się dużej liczby wirusów i bakterii mogą to być także urzędy, muzea, stadiony czy sale koncertowe. Zwłaszcza zaś te miejsca, w których ludzie często korzystają z tych samych urządzeń np. z dotykowymi ekranami jak choćby kioski informacyjne w centrach handlowych, punkty poboru biletów (w urzędach czy w laboratoriach) czy ekrany multimedialne w muzeach. Użytkownicy takich systemów w miejscach często odwiedzanych przez ludzi są bowiem bardziej narażeni na zarażenie grzybem, bakterią czy wirusem co, wobec masowości wizytujących użytkowników, może szybko przekształcić się w epidemię o zasięgu co najmniej lokalnym. Prewencja zarażeniem przez unikanie dotyku w zbędnych sytuacjach powinna być szczególnie ważna w dobie rozwoju bakterii odpornych na antybiotyki [18][19][20][21].

Zaprezentowane zalety i dodatkowe możliwości wynikające z bezdotykowej obsługi szczególnie urządzeń mobilnych takich jak okulary multimedialne czy gogle VR bazujące na telefonach wskazują, iż rozwój tego typu interakcji jest dziedziną atrakcyjną i o znacznej wadze. Z tego powodu, w niniejszej rozprawie doktorskiej podjęta została tematyka projektu i budowy bezdotykowego czujnika gestów dłoni z uwzględnieniem do zastosowania w urządzeniach mobilnych odznaczającego się niskim poborem energii i szeroką bazą rozpoznawanych gestów.

1.3. Taksonomie gestów

Budowa czujnika gestów i projekt metod pozwalających na ich rozpoznanie powinny być poprzedzone usystematyzowaniem samego pojęcia gestu i analizą zaproponowanych podziałów gestów na rodzaje i kategorie celem odniesienia możliwości projektowanego czujnika względem prac innych autorów.

Gest według Kendona to: „*wszelkie wyraźne działania ciała, które są uważane przez uczestników jako bezpośrednio zaangażowane w proces zamierzonej wypowiedzi*” [22]. Podobnie gest formułuje Ishino jako „*widoczne i intencjonalne akcje wykonywane za pomocą ciała przedstawiające sobą określone znaczenie*” [23]. Voylesa i in. opisali gest jako

„nieprecyzyjne, zależne od kontekstu zdarzenie, które przekazuje intencje osoby” [24]. Z definicji tych wynika, iż gesty mogą być zarówno mimiką twarzy jak i ruchem nogi, ręki czy gałki ocznej. W kontekście niniejszej rozprawy zakres ten zostanie jednak zawężony do analizy ruchu i pozy dłoni. Z przytoczonych definicji można także wyciągnąć wniosek, iż dokładnie ten sam gest, lecz rozpatrywany w różnym kontekście, może nieść ze sobą inne znaczenie. Przykładem takiej sytuacji podczas obsługi programów komputerowych może być wciśnięcie na klawiaturze klawisza F11 powodującego wyzwolenie innego typu akcji w aplikacji edytora tekstowego (przeskok do następnego pola) oraz inne działanie w przeglądarce internetowej (załączanie trybu pełnego ekranu). W przypadku obsługi interfejsu graficznego użytkownika za pomocą gestów również dopuszczalne jest aby, w zależności od kontekstu (np. rodzaju wyświetlanego pliku: tekstowego lub multimedialnego) różne akcje wyzwalane były tym samym gestem. Szczególnie istotne jest natomiast aby rodzaj wykonywanego gestu intuicyjnie wskazywał użytkownikowi na rodzaj wyzwalanej przez niego akcji (np. przesunięcie dłonią spowoduje przesunięcie podświetlenia aktywnej ikony w GUI, dłoń statyczna włączy opcję itp.).

Gesty, jako środek komunikacji międzyludzkiej ale także między człowiekiem a maszyną, poddawane były na przestrzeni lat różnym podziałom ze względu na rodzaje, kontekst czy kategorie. W literaturze znaleźć można zróżnicowane klasyfikacje czyli taksonomie gestów.

W studium dotyczącym spostrzeżeń na temat historii gestów, lecz bez rozpatrywania ich interpretacji przez maszynę, Kendon zwraca szczególną uwagę na gesty pantomimiczne, emblematyczne, gestykulacyjne i język znakowy [22]. W dziedzinie rozpoznawania gestów ręki na podstawie obrazu wideo Wu i Huang wyróżnili cztery kategorie gestów: komunikacyjne, kontrolne, manipulacyjne i konwersacyjne [25]. Celem gestów komunikacyjnych jest przekazanie konkretnej informacji. Używany przez osoby głuchonieme język migowy, z założenia mocno ustrukturyzowany i precyzyjny, reprezentuje taki właśnie typ gestów. W kontekście interakcji z interfejsem graficznym użytkownika do wskazywania obiektów (ikon) oraz nawigowania między nimi użytkownik korzystać będzie z gestów kontrolnych. Gesty manipulacyjne z kolei sprowadzają się do interakcji z wirtualnymi obiektami widzianymi przez użytkownika, czyli do oddziaływania swoją aktywnością na ich zachowanie czy pozycję (np. przesuwanie gestem wirtualnej piłki względem innych obiektów wyświetlanych w GUI). Gesty konwersacyjne są z reguły spontaniczne

i niezaplanowane, nie stanowią osobnej informacji lecz jedynie towarzyszą wiadomości przekazywanej za pomocą mowy [26]. Z perspektywy przydatności poszczególnych z opisanych kategorii gestów w kontekście obsługi urządzeń, prezentowane gesty komunikacyjne mogą być przeznaczone do wyzwalania konkretnych akcji systemu. Gesty kontrolne i manipulacyjne dostarczają funkcjonalności zapewniającej nawigację i interakcję. Z kolei gesty konwersacyjne nie powinny być w ogóle uwzględniane jako komendy sterujące.

Klasyfikacja sposobów interakcji człowieka z komputerem bazujących na gestach została ujęta w odniesieniu do czterech dziedzin przez Karam i Schraefela [27]:

- obszar zastosowań (np. gry, urządzenia mobilne, rozszerzona rzeczywistość, operacje chirurgiczne, itp.),
- wykorzystywane technologie (np. klawiatura, myszka, inteligentne rękawiczki, czujnik optyczny, itp.),
- sposób reakcji komputera (np. przerwanie sprzętowe, komunikat głosowy, przerwanie programowe – zdarzenie, komunikat graficzny, itp.),
- rodzaj (styl) gestu (np. wskazujący ruch dłoni, symbol języka migowego, powiązanie ruchu dłoni z obsługą manipulatora, itp.).

Ze względu na zakres niniejszej rozprawy doktorskiej szczególnie interesujący jest podział rodzajów (stylów) gestów związanych z dłonią. Na podstawie rozważań Queka [28] i przeprowadzonego przeglądu Karam i Schraefel dokonali rozbicia dziedziny rodzaje gestów na gesty deiktyczne (wskazujące, ofensywne), manipulacyjne (np. imitacja ruchu pokrętle), symboliczne (np. poza dłoni o określonym znaczeniu – kciuk w górę), gestykulacyjne (towarzyszące mowie przez wzbogacenie przekazu słownego, podobnie jak konwersacyjne [29]) i gesty o znaczeniu językowym (np. język migowy) czy kombinacje powyższych.

Gesty manipulacyjne, według Queka, to akcje, w których intencją jest kontrola wirtualnego obiektu przez ścisłą zależność między ruchami ręki wykonującymi gest a ruchami obiektu w manipulowanej przestrzeni wirtualnej. Przykładem takich gestów, w kontekście interfejsu człowiek-komputer, może być odwzorowywanie stopniowej zmiany położenia dłoni względem czujnika gestów (np. z prawej na jego lewą stronę, stopniowe oddalanie lub zbliżanie dłoni do czujnika) w przemieszczanie się w analogicznym kierunku samolotu na ekranie w trakcie gry (np. z prawej na lewą część ekranu, z dolnej na górną).

Gesty symboliczne wykorzystywać mogą pozy i gesty (głównie dynamiczne) jako elementy słownika, które traktowane są przez system jako symbole pozwalające na komunikację z maszyną przez wydawanie prostych poleceń, np. włącz, cofnij itp. (stąd nazywane są też gestami komunikacyjnymi). Pozą statyczną dłoni nazywać będziemy zaś ułożenie palców dłoni niezmiennie się w czasie i przestrzeni. W dalszej części rozprawy skrótowe oznaczenie „poza” odnosić się będzie do pozy statycznej. Pozą dynamiczną dłoni będziemy nazywali zmieniające się ułożenia palców dłoni pozostającej w jednej lokalizacji. Gestem dynamicznym będzie nazywana ekspresja, w której zmienia się przede wszystkim położenie dłoni w trakcie wykonywania gestu (np. przesunięcie dłoni wzdłuż czujnika gestów).

Taksonomię, która precyzyjnie grupuje gesty wykonywane na powierzchni ekranu dotykowego zaprezentowali Wobbrock i in. [30]. Autorzy klasyfikowali gesty wykorzystując cztery wymiary: formę, naturę, powiązanie i płynność (ang. *form, nature, binding, flow*), z których każdy składa się z kilku kategorii (Tabela 1.1).

Tabela 1.1 Taksonomia gestów według czterech wymiarów Wobbrocka.

Wymiar gestu	Kategorie wymiaru gestu	Opis gestu
forma (ang. <i>form</i>)	Poza statyczna (ang. <i>static pose</i>)	Poza dłoni niezmienna i w jednej lokalizacji
	Poza dynamiczna (ang. <i>dynamic pose</i>)	Poza dłoni zmienna i w jednej lokalizacji
	Poza statyczna i ścieżka (ang. <i>static pose and path</i>)	Poza dłoni niezmienna i dłoń przemieszcza się
	Poza dynamiczna i ścieżka (ang. <i>dynamic pose and path</i>)	Poza dłoni zmienna i dłoń przemieszcza się
	Jednopunktowy dotyk (ang. <i>one-point touch</i>)	Poza statyczna z jednym palcem
	Jednopunktowa ścieżka (ang. <i>one-point path</i>)	Poza statyczna z jednym palcem i ścieżka
natura (ang. <i>nature</i>)	Symboliczna (ang. <i>symbolic</i>)	Gest wizualnie obrazuje symbol
	Fizyczna (ang. <i>physical</i>)	Gest działa fizycznie na obiekt
	Metaforyczna (ang. <i>metaphorical</i>)	Gest wskazuje metaforę
	Abstrakcyjna (ang. <i>abstract</i>)	Mapowanie gestów-referentów jest arbitralne
powiązanie (ang. <i>binding</i>)	Z obiektem (ang. <i>object-centric</i>)	Lokalizacja względem cech obiektu
	Zależne od świata (ang. <i>world-dependent</i>)	Lokalizacja względem cech świata
	Niezależne od świata (ang. <i>world-independent</i>)	Lokalizacja ignoruje cechy świata
	Mieszane zależności (ang. <i>mixed dependencies</i>)	Niezależne od świata oraz inne
płynność (ang. <i>flow</i>)	Dyskretna (ang. <i>discrete</i>)	Odpowiedź następuje po akcji użytkownika
	Ciągła (ang. <i>continuous</i>)	Odpowiedź następuje podczas akcji użytkownika

Podział ten został utworzony po analizie wyników doświadczenia motywowanego jedną z zasad użyteczności projektowania ukierunkowanego na użytkownika (ang. *user centered design*). W badaniu eksperymentalnym poproszono bowiem ochotników aby sami zdecydowali o sposobie obsługi systemu. Grupie użytkowników zostały początkowo wyświetlone na interaktywnym stole różne reakcje systemu operacyjnego (np. pomniejszające się zdjęcie), którym towarzyszył ich odczytywany opis słowny (np. „zmniejszające się zdjęcie”). Następnie ochotnicy byli proszeni o wykonanie na dotykowym panelu gestu, który ich zdaniem w najbardziej naturalny sposób, mógłby wyzwolić zaprezentowane zdarzenie. Przeprowadzone badania pozwoliły na opracowanie podziału gestów nie tylko ze względu na cel czy kontekst gestu lecz także ze względu na charakter ruchu i położenia dłoni (czy poza się zmieniała, czy dłoń zakreślała trajektorię, czy ważna jest lokalizacja dłoni itp.).

Na podstawie dokonanego przeglądu taksonomii zwrócono uwagę na następujące czynniki podziału gestów. Jednym z nich jest płynność gestu (wymiar *flow* Wobbrocka) [30] składająca się z dwóch kategorii. Pierwsza z nich, gesty dyskretne, oznacza, iż reakcja systemu następuje dopiero po obserwacji całego gestu, czyli musi się on zakończyć (np. naciśnięcie przycisku myszy komputerowej). Rozpoznanie gestów dyskretnych może całkowicie odbywać się w obrębie czujnika gestów i na klasyfikacji gestów tego typu skupiono się w niniejszej rozprawie doktorskiej. Obsługa gestów drugiej kategorii, gestów ciągłych, odbywa się zaś w trakcie wykonywania gestu i polega na śledzeniu trajektorii ruchu użytkownika i reagowaniu na niego na bieżąco przez system (np. przesuwanie kursora po ekranie wraz z ruchami myszy na biurku). Drugim z czynników podziału gestu, na który zwrócono uwagę jest rozróżnienie na gesty tworzone przez pozy statyczne (nieruchoma dłoń z konkretnym ułożeniem palców znajdująca się przed czujnikiem) oraz statyczne pozy o dynamicznej ścieżce (dłoń z konkretnym ułożeniem palców przemieszczająca się np. wzdłuż czujnika) (wymiar *form* Wobbrocka) [30]. Gesty ostatniego typu, ze ścieżką ruchu równoległą lub prostopadłą do czujnika gestów, będą w dalszej części pracy nazywane przesunięciem dłoni.

Ponadto istotne będzie zaproponowanie takich gestów dyskretnych rozpoznawanych przez opracowywany czujnik gestów, które będą mogły być zaliczane do kategorii gestów kontrolnych lub symbolicznych (np. przesunięcie dłoni w lewo / w prawo, pokazanie pozy z jednym / dwoma palcami). Dzięki takiemu rozróżnieniu gesty przypisane do konkretnych

reakcji graficznego interfejsu użytkownika będą bardziej intuicyjne w swoim znaczeniu. W kontekście gestów manipulacyjnych, które są w dużej mierze typu ciągłego, w niniejszej rozprawie badane będą możliwości obsługi tego typu działań przez czujnik optyczny.

1.4. Wymagania użytkowe i techniczne interfejsu

Podjmując zagadnienie opracowania metod bezdotykowej interakcji z użyciem optycznego czujnika gestów należy zwrócić uwagę na szereg wymagań zdefiniowanych dotychczas w literaturze dla tego typu urządzeń.

W odniesieniu do interfejsów wizyjnych Murthy wskazał na cztery wymagania: odporność, efektywność obliczeniowa, tolerancja użytkownika i skalowalność (ang. *robustness*, *computational efficiency*, *user's tolerance*, *scalability*) [31]. Kontekst przytoczonej publikacji wskazuje, iż wymagania dotyczyły obrazu wideo, jednak są one na tyle uniwersalne, iż można je zaadaptować do interfejsów innych typów. *Odporność* związana jest z nieczułością sensora na zmienne warunki użytkowania. Powinien on zatem cechować się możliwością adaptacji do zróżnicowanych sytuacji tak aby jego dokładność działania pozostawała na wysokim poziomie dla możliwie największej różnorodności warunków pracy. *Efektywność obliczeniowa* z kolei jest szczególnie ważna w przypadku wieloprotocowych aplikacji działających w czasie rzeczywistym na platformie mobilnej. Mało złożony czujnik często dostarcza danych, których przetworzenie jest mniej wymagające obliczeniowo niż np. w przypadku sekwencji wideo. *Tolerancja użytkownika* to przede wszystkim odporność systemu na interpretację omyłkowo wykonanych ruchów, nie będącymi gestami. *Tolerancja* może być także związana z cechami osobniczymi jak z wielkością dłoni lub z poczuciem prędkości i odległości od czujnika przez osobę wykonującą gest. *Skalowalność* mówi o tym, że system interakcji powinien być adaptowalny do zróżnicowanych zastosowań (np. w obsłudze urządzeń o interfejsach graficznych użytkownika o różnym stopniu złożoności).

Ponadto Cheng i in. zwracają również uwagę na następujące wymagania: rozpoznawanie w czasie rzeczywistym, brak konieczności uczenia modelu celem rozpoznawania gestów nowych użytkowników, minimalizacja fałszywych alarmów i rozpoznawanie fazy rozpoczęcia i zakończenia gestu [32]. Projektowanie sposobów detekcji wykonywanych gestów jest często kompromisem pomiędzy minimalizacją poboru energii przez czujnik poprzez okresowe wprowadzanie go w stan uśpienia a zachowaniem jego zdolności do wykrywania gestów budzących go z tego stanu. Reakcja systemu na wykonany

gest w czasie możliwie zbliżonym do zera zapewnia wygodną dla użytkownika obsługę systemu. Z tego powodu należy zadbać zarówno o krótki czas przetwarzania danych jak i optymalizację szybkiego wybudzania czujnika ze stanu uśpienia. Sensor rozpoznający gesty z jednakową dokładnością niezależnie od użytkownika, nie wymagający mechanizmu kalibracji, stanowi zaś rozwiązanie najbardziej wygodne i praktyczne. Wykonanie przez system niechcianej komendy jest z kolei bardziej niepożądane niż całkowity brak reakcji na dany gest, który zawsze może zostać powtórzony. Na potrzebę zastosowania podziału gestu na fazy, wyznaczone przez (1) podjęcie aktywności, (2) wykonanie gestu oraz (3) powrót do spoczynku (np. przez rękę) uwagę zwrócił z kolei także Yin [33].

Odnosząc się do wymagań użytkowych, zdaniem autora niniejszej rozprawy, cztery gesty wydają się być absolutnym minimum jeśli chodzi o ilość gestów rozpoznawanych przez interfejs dla urządzeń mobilnych. Dzięki nim użytkownik będzie w stanie (1) zaakceptować akcję, (2) anulować ją (lub cofnąć) oraz przemieszczać się (3) ku następnemu lub (4) poprzedniemu elementowi odpowiednio zaprojektowanego graficznego interfejsu użytkownika. Z punktu widzenia użytkowego przy analizie możliwości danego czujnika gestów, należy zwrócić uwagę na ilość podstawowych póz statycznych, które jest on w stanie rozróżnić. Na ich podstawie można bowiem, korzystając z odpowiedniego modelu, budować sekwencje póz składających się na gesty złożone powiększając w ten sposób podstawową kolekcję rozpoznawanych akcji. Rozpoznawanie gestów dynamicznych z kolei pozwoli na przypisanie im wyzwalania bardziej adekwatnych reakcji systemu, ze względu na dynamikę ruchu (np. przełączanie aktywnej ikonki). Większa kolekcja gestów oznacza zaś szybsze wyzwalanie konkretnych akcji co znacząco podnosi komfort użytkownika.

W niniejszej rozprawie doktorskiej szczególnie ważnym zastosowaniem dla projektowanego w ramach rozprawy interfejsu są okulary multimedialne. Opracowywany czujnik gestów mógłby być ukryty w obrębie oprawek takiego urządzenia. Z tego powodu, wielkość czujnika nie musi być za wszelką cenę minimalizowana, zwłaszcza jeśli to dzięki niej będzie on mógł osiągnąć pewne przewagi nad rozwiązaniami innego typu. Czujnik musi być jednak umieszczony w oprawce w taki sposób by sterowanie za jego pomocą było możliwie wygodne i naturalne dla osoby obsługującej okulary.

1.5. Cel, tezy i zakres pracy

Tematyka niniejszej rozprawy doktorskiej dotyczy eksploracji możliwości bezdotykowego, optycznego czujnika gestów dłoni o budowie liniowej i jego praktycznego wykorzystania szczególnie w urządzeniach mobilnych.

Celem głównym pracy doktorskiej jest *zaprojektowanie metod rozpoznawania gestów dłoni i sposobów interakcji z urządzeniami za pomocą kilkudetektorowego optycznego czujnika o liniowej budowie.*

W ramach problemów badawczych zdefiniowano następujące tezy:

- I. *Możliwe jest opracowanie kilkuelementowego czujnika optycznego zbudowanego z liniowo rozłożonych fotodiod oraz diod LED, jak również przesłon optycznych formujących pole widzenia fotodiod, umożliwiającego rozpoznawanie co najmniej 4 póż dłoni.*
- II. *Opracowanie dwóch trybów pracy optycznego czujnika o liniowej budowie z zastosowaniem (tryb aktywny) lub bez zastosowania (tryb pasywny) dodatkowego oświetlenia w podczerwieni oraz opracowanie dedykowanych im metod przetwarzania danych umożliwia uzyskanie wyższej niż 90% dokładności rozpoznawania póż dłoni przy minimalizacji energii dla potrzeb zastosowań mobilnych.*
- III. *Opracowanie modelu bazującego na rekurencyjnej sieci neuronowej wykorzystującego jako dane wejściowe nieprzetworzone pakiety danych z optycznego czujnika o liniowej budowie pozwala na klasyfikację co najmniej 20 gestów dłoni z dokładnością nie mniejszą niż przy wykorzystaniu danych przetworzonych w formie wektora cech z rozpoznanymi pozami dłoni.*

Pierwsza z tez dotyczy weryfikacji możliwości projektowanego kilkuelementowego czujnika o liniowej budowie w rozróżnianiu (statycznych) póż dłoni. Zapewnienie rozróżniania kilku póż, biorąc pod uwagę również inne stopnie swobody czujnika (przemieszczenie, lokalizacja), umożliwi sformułowanie szerokiej bazy gestów.

Druga teza związana jest z maksymalizacją dokładności klasyfikacji póż i efektywności energetycznej optycznego czujnika gestów możliwie niezależnie od warunków świetlnych otoczenia. Jak w przypadku czujników optycznych, zasada działania

rozpatrywanego czujnika, w zależności od trybu pracy, jest z założenia podatna zarówno na silne światło otoczenia jak i pomijalny jego poziom.

Trzecia teza odnosi się do rodzaju danych jaki należy podać klasyfikatorowi gestów aby zmaksymalizować jego dokładność rozpoznawania gestów. Ponadto zawiera ona deklarację liczby rozpoznawanych gestów (20), wyraźnie większej niż w przypadku innych podobnego typu czujników opisane w literaturze.

W ramach podjęcia prac nad zweryfikowaniem tez podzielono zakres zadań niniejszej rozprawy doktorskiej na cele szczegółowe:

- A. zdefiniowanie liczby gestów wymaganej do wygodnej, według przyjętych założeń, obsługi urządzeń, w szczególności urządzeń mobilnych,
- B. zaprojektowanie i realizacja kilkudetektorowego optycznego czujnika gestów o liniowej budowie umożliwiającego interakcję bezdotykową,
- C. weryfikację właściwości sensora optycznego poprzez przeprowadzenie badań symulacyjnych i eksperymentalnych z użyciem fizycznej implementacji czujnika,
- D. opracowanie metod klasyfikacji póż i gestów dłoni dedykowanych dla optycznego czujnika o budowie liniowej,
- E. weryfikację działania czujnika optycznego w różnych warunkach oświetlenia i rozważenie różnych trybów pracy,
- F. praktyczne zastosowanie czujnika gestów w różnych rozwiązaniach konstrukcyjnych.

1.6. Układ pracy

Całość materiału poznawczego i badawczego została podzielona na pięć części, każda zawarta w osobnym rozdziale. Najważniejsze zagadnienia poruszane w poszczególnych rozdziałach niniejszej rozprawy doktorskiej zostały zaprezentowane na schemacie (Rys. 1.1).

Rozdział 1: Wprowadzenie	Rozdział 2: Czujniki i metody	Rozdział 3: Optyczny czujnik gestów	Rozdział 4: Klasyfikacja póż i gestów dłoni	Rozdział 5: Zastosowanie i podsumowanie
Rola bezdotykowych interfejsów	Typy czujników gestów: optyczne + pozostałe	Podstawy koncepcyjne i zasada działania	Schemat systemu rozpoznawania gestów	Podsumowanie opracowanych rozwiązań
Taksonomie gestów	Metody klasyfikacji jednej obserwacji i sekwencji	Parametryzacja i wymagania dla czujnika	Klasyfikacja póż	Testy interakcji w różnych rozwiązaniach konstrukcyjnych
Wymagania użytkowe i techniczne	Miary efektywności metod klasyfikacji	Metodologia ewaluacji: symulacje + pomiary	Wpływ zmiennych warunków pracy na właściwości czujnika	Przyszłe prace i praktyczne wykorzystanie czujnika
Cel i tezy pracy	Projekt zbioru akcji do obsługi urządzeń mobilnych	Weryfikacja właściwości czujnika optycznego	Klasyfikacja gestów	Podsumowanie rozprawy doktorskiej

Rys. 1.1 Schemat układu rozprawy doktorskiej wraz z kluczowymi zagadnieniami każdego z rozdziałów.

Niniejszy rozdział 1 stanowi wprowadzenie do tematu. W tej części zarysowano ewolucję urządzeń mobilnych od pagerów do postaci ubieralnej elektroniki w formie multimedialnych okularów. Wskazano również na możliwe pola zastosowań okularów multimedialnych ze szczególnym uwzględnieniem wykorzystania bezdotykowych interfejsów. Podjęto problem taksonomii gestów i odniesiono go do zdefiniowanych dotychczas w literaturze propozycji. Dokonano analizy wymagań użytkowych i technicznych interfejsów stosowanych w urządzeniach mobilnych. Zaprezentowano i wyjaśniono także trzy tezy, tworzące problemy badawcze niniejszej rozprawy doktorskiej oraz jej cel.

Rozdział 2 przedstawia przegląd literatury, podsumowujący możliwości wybranych interfejsów gestów. Szczegółowo omówione zostały czujniki optyczne z uwzględnieniem podziału na ilość detektorów (jeden, kilka, wiele) oraz poglądowo sensory bazujące na pobudzeniach innego typu. W dalszej części omówiono metody klasyfikacji możliwe do wykorzystywania przy rozpoznawaniu póż i gestów dłoni.

Podstawy koncepcyjne i fizyczna konstrukcja optycznego sensora gestów opisane zostały w rozdziale 3. Celem weryfikacji właściwości czujnika, przeprowadzono szereg badań symulacyjnych działania optycznego sensora opartych na metodzie Monte Carlo.

W dalszej części rozdziału porównano charakterystyki sensora w postaci wyników symulacji i pomiarów.

W rozdziale 4 omówiono szczegółową koncepcję i ewaluację wybranych metod rozpoznawania pól dłoni. Opisano również przeprowadzone badania dotyczące wpływu różnych warunków oświetlenia na pracę czujnika i rozważono różne jego tryby pracy. Rozdział kończy analiza dokładności rozpoznawania szerokiej bazy gestów przez zastosowanie jednego z modeli do klasyfikacji sekwencji.

W ostatniej części rozprawy, w rozdziale 5, zaprezentowano praktyczne przykłady użycia czujnika gestów, możliwości rozwoju użyteczności koncepcji bezdotykowego liniowego sensora gestów o liniowej budowie oraz przedstawiono dyskusję i podsumowanie przeprowadzonych prac i wyników.

2. CZUJNIKI I METODY ROZPOZNAWANIA GESTÓW

Niniejszy rozdział rozpoczyna omówienie rozwiązań czujników gestów dłoni zaprezentowanych w literaturze gdzie ze szczególną uwagą, ze względu na tematykę niniejszej rozprawy, przedstawiono czujniki optyczne. Dla sensorów optycznych przedstawiono podział ze względu na ilość elementów rejestrujących zmiany w otoczeniu urządzenia oraz, dla pozostałych czujników, ze względu na rodzaj pobudzenia rejestrowanego przez przetwornik. Następnie zaprezentowano opis schematu przedstawiającego procesy jakie zachodzą w klasycznych systemach rozpoznawania gestów, niezależnie od stopnia ich złożoności. Kolejna sekcja składa się z dwóch części. Pierwsza z nich dotyczy opisu wybranych metod klasyfikacji danych stacjonarnych, które rozważono przy wyborze modelu rozpoznawania pozy dłoni na podstawie sygnału z proponowanego czujnika o liniowej budowie. Druga część odnosi się do metod klasyfikacji całych sekwencji trwających w czasie, za pomocą których możliwe jest rozpoznawanie gestów, uwzględniając potencjalną zmianę pozy czy lokalizacji ręki w trakcie wykonywania gestu. Następnie opisano ważniejsze ze stosowanych w rozprawie miar efektywności metod klasyfikacji. W dalszej części rozdziału zdefiniowany został podstawowy zbiór rozważanych w pracy poz dłoni. Ostatnia część rozdziału podejmuje problematykę określenia ilości akcji potrzebnych do precyzyjnej i sprawnej nawigacji urządzenia mobilnego, która determinować będzie zapotrzebowanie na ilość różnych gestów rozpoznawanych przez czujnik gestów dla urządzeń przenośnych.

2.1. Typy czujników gestów

Berman i Stern przedstawili taksonomię sensorów gestów ze względu na ich integrację z systemem rozpoznawania gestów [34]. Zaproponowali oni podział ze względu na: rodzaj pobudzenia sensora, kontekst użycia i platformę sensora. W ramach zróżnicowania, co do sposobu pobudzenia wyróżniono 5 grup sensorów: elektryczne, optyczne, akustyczne, magnetyczne i mechaniczne. W pierwszej części przeglądu rozwiązań obecnych w literaturze omówiono przykłady bezkontaktowych czujników optycznych, gdyż są one związane z tematyką niniejszej rozprawy. W dalszej części poglądowo przedstawiono wybrane bezdotykowe rozwiązania detekcji gestów dłoni bazujące na sensorach pobudzanych w sposób inny niż optycznie.

2.1.1. Optyczne czujniki gestów

Sensory optyczne można rozważyć w kategoriach zróżnicowanych ze względu na ilość elementów rejestrujących (detektorów). Ponadto, niezależnie od tego kryterium podziału wyróżnić można typ każdego czujnika: pasywny (niewymagający pobudzenia) lub aktywny (bazujący na stosowaniu pobudzenia).

a) Czujniki jedno- i dwudetektorowe

Obszerną grupę sensorów optycznych stanowią czujniki posiadające jeden detektor poziomu natężenia światła otoczenia (najczęściej jest to fotodioda lub fototranzystor). Jednodetektorowe pasywne czujniki optyczne, pozbawione własnego systemu podświetlenia, wprowadzone do masowej produkcji często z założenia pełnią rolę czujników światła otoczenia (ang. *ambient light sensor*). Wśród komercyjnych rozwiązań tego typu wyróżnić można czujniki OPT3001, NOA1305 czy APDS-9009 [35][36][37]. Możliwe jest jednak użycie tego typu przetworników jako sensorów zbliżeniowych (ang. *proximity sensor*). Zasada działania pasywnych czujników zbliżeniowych polega na tym, że (w obecności jasnego światła otoczenia) obiekt pojawiający się w ich bliskim polu widzenia, przysłania światło otoczenia padające na powierzchnię czujnika. Spadek sygnału na elemencie światłoczułym (niższy poziom intensywności rejestrowanego światła) można w kontekście rozpoznawania gestów w uproszczeniu potraktować jako pojawienie się dłoni. Zastosowania tego typu rozwiązań można znaleźć np. w smartfonach gdzie informacja z takiego czujnika pozwala na wyłączenie ekranu po przyłożeniu telefonu do ucha. Pasywne jednodetektorowe czujniki światła otoczenia zamieszczone np. w multimedialnych

zegarkach (ang. *smart watch*) pozwalają na monitorowanie aktywności człowieka przy komputerze (częstotliwość wciskania klawiszy klawiatury) [38].

Wiele czujników zbliżeniowych pracuje jednak w trybie aktywnym, czyli z zastosowaniem dodatkowego źródła światła. Jednym z takich przykładów jest rodzina elementów Sharp GP2Y. Przykładowy czujnik, GP2Y0A21YK, składa się z jednej pary fotodiody - dioda LED [39]. Mierzy on intensywność światła odbitego od obiektu, emitowanego pulsacyjnie przez diodę LED, która jest proporcjonalna do dystansu między czujnikiem a obiektem. Pulsacja jest stosowana celem redukcji zużycia energii. Nie ma bowiem potrzeby aby diody LED zapalone były przez cały czas a jedynie na okres potrzebny do przeprowadzenia próbkowania sygnału odbitego. Urządzenie GP2Y0A21YK potrafi określić odległość do obiektów znajdujących się w przedziale od 10 do 80 cm od sensora. Typowe zużycie prądu przez czujnik wynosi 30 mA przy napięciu zasilania 4 V. Częstotliwość podświetlenia obiektu i próbkowania sygnału wynosi ok. 25Hz. Podobną zasadą działania cechują się czujniki Si1102 firmy Silicon Labs [40], VCNL3020 firmy Vishay Semiconductors [41] czy HSDL 9100 produkcji Avago Technologies [42]. Zaletą czujników aktywnych jest większa niezależność od warunków świetlnych otoczenia (częściej występuje niski niż zbyt wysoki poziom natężenia światła) czy większa dokładność oszacowania odległości od obiektu odbijającego światło jednak kosztem większego zapotrzebowania na zużycie energii.

Wśród szeregu interfejsów, zbliżeniowy czujnik aktywny używany był w prototypowym urządzeniu wykorzystywanym do jednego z wcześniejszych badań nad bezkontaktową interakcją z urządzeniami mobilnymi w roku 2000 [43]. Za pomocą czujników jednodetektorowych (GP2Y0D340K) możliwe jest także rozróżnienie niektórych układów ułożenia palców. Wymaga to jednak ruchu dłonią prostopadle do czujnika i w bliskiej do niego odległości, aby był on w stanie rozróżnić sygnały odbite od poszczególnych rozpostartych palców [44]. Sensor taki zlicza ilość przesunięć wykonanych przez dłoń przed urządzeniem. Czujnik jest z założenia umieszczony koło ucha, toteż gesty są wykonywane blisko głowy, zatem jest on potencjalnie łatwo integrowalny z urządzeniami typu okulary multimedialne.

Czujniki aktywne z tylko jednym elementem światłoczułym umożliwiają rozróżnienie kierunku ruchu obiektu w jednej osi w płaszczyźnie powierzchni aktywnej sensora. Rozwiązania, których działanie jest możliwe przy użyciu również tylko jednej diody LED

bazują na gradiencie czułości fotodiody [45], pochłaniaczu światła [46] bądź idei asymetrycznych bloków (przesłon) optycznych [47].

System z dwiema diodami LED powoduje, iż niepotrzebne stają się dodatkowe elementy modyfikujące wiązkę światła dla celu określenia kierunku ruchu. Przy takiej konstrukcji można obliczyć fazy sygnału natężenia światła odbitego od przesuwanej się dłoni. W tej koncepcji oświetlenie odbywa się naprzemiennie przez dwie diody LED zaś rejestracja przez fotodiode znajdującą się pomiędzy źródłami światła. Metoda wymaga ułożenia od siebie diod LED w odpowiednio dużej odległości, w stosunku do częstotliwości próbkowania sygnałów. Warunkiem jest, aby w zakresie prędkości ruchów ludzkiej dłoni możliwe było obliczenie zauważalnej różnicy między sygnałami natężenia światła odbitego od obiektu oświetlanego pod różnymi kątami. Naprzemiennie zapalanie dwóch oddalonych od siebie diod LED i próbkowanie ich sygnałów do osobnych kanałów pozwala na zauważenie wzajemnego przesunięcia obydwu przebiegów, którego kierunek jest zdeterminowany kierunkiem ruchu dłoni. Detekcja kierunku ruchu bazująca na tej metodzie została wykorzystana m.in. w [32][48][49][50].

Jednym z wymienionych czujników tego typu jest odbiciowy sensor gestów opisany przez Chenga i in. [32](Rys. 2.1a). W publikacji nie podana została jednak informacja o odległościach między elementami optoelektronicznymi tudzież czy wykorzystane zostały bloki optyczne. Zastosowana częstotliwość próbkowania wynosi 100Hz zaś szerokość pojedynczej ramki przetwarzanych danych równa się 14 próbkom, zatem czas akwizycji danych wynosi 140ms. Przygotowując dane do analizy wykorzystano okno ramki, w efekcie którego ramka próbkowana w poprzednim cyklu i obecnym nakładają się w 50% tworząc analizowany bufor. Każda ramka, która spełni odpowiednie wymagania (odpowiedni poziom wariancji wartości próbek w ramce danych) jest następnie poddawana ekstrakcji cech, na podstawie których określany jest rodzaj wykonanego gestu.

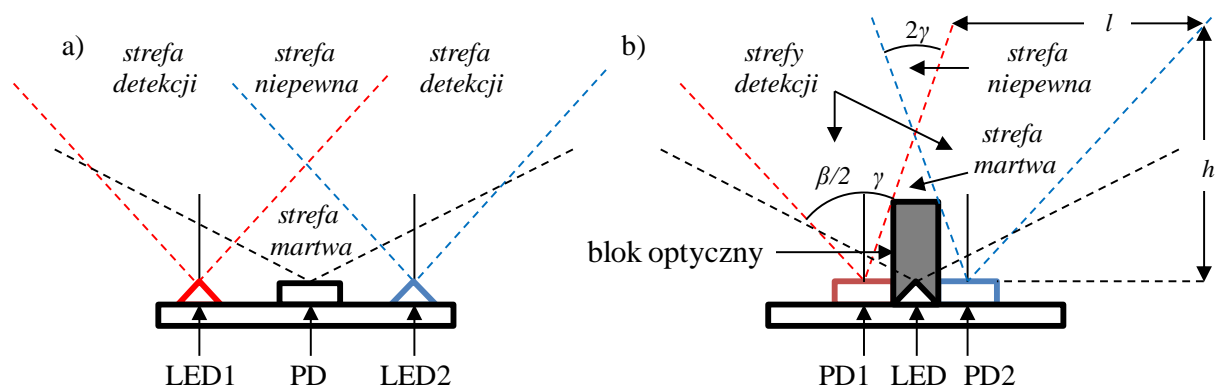
Pierwszą z cech jest *opóźnienie międzykanałowe*. Wyraża się je przez czas opóźnienia, t_D , obliczany przez znalezienie liczby ramek n , która wskazuje jak mocno sygnały z obydwu kanałów przesunięte są względem siebie. Miarą minimalnego przesunięcia jest największa wartość funkcji korelacji wzajemnej dwóch sygnałów dyskretnych (z obydwu kanałów). Kolejną cechą jest *lokalna suma zboczy*, która opisuje lokalne zbocze sygnału w obrębie ramek danych. Zbocze w pojedynczej ramce danych wyznaczane jest na podstawie regresji liniowej pierwszego rzędu a następnie sumowane ze zboczami z sześciu poprzednich

ramek (aby lepiej oddać trend ruchu). Znak współczynnika funkcji regresji liniowej wskazuje na kierunek ruchu ręki względem czujnika w osi równoległej do normalnej do sensora. Wartość tego współczynnika mówi o szybkości wykrytego ruchu (im ruch szybszy, tym większa wartość tego parametru). Pozostałe cechy związane są ze statystycznym opisem sygnału. Obliczana jest wartość średnia oraz wariancja ramki danych z sensora. Przyjęto, iż duża wariancja oznacza zaobserwowanie gestu w polu widzenia sensora, niska zaś wskazuje na brak ruchu ręki w strefie widzianej przez sensor. Po zakończeniu procesu ekstrakcji cech wyniki przekazywane są jako argumenty do drzewa decyzyjnego.

Na podstawie cech z aktualnie przetworzonej oraz siedmiu poprzednich ramek klasyfikator podejmuje decyzję czy i jaki gest został zaobserwowany. Sensor jest w stanie rozpoznać kierunek ruchu dłonią (w prawo lub lewo i od lub do sensora) z dokładnością malejącą wraz z rosnącym dystansem między dłonią a czujnikiem h . Wynosi ona ok. 80% dla $h=3$ cale. Dokładność zastosowanych metod rozpoznawania gestów jest większa niż 90% dla gestów wykonywanych w tempie od 2 do 4 na sekundę. Eksperyment ten wykonany został z pomocą elektronicznego metronomu wydającego dźwięki w zadanym tempie, które ochotnicy mieli za zadanie odzwierciedlić okresowością pokazywanych gestów. W optymalnej odległości i kiedy gesty wykonywane są z optymalną prędkością, precyzja metody wyniosła 98%. Niższa wartość czułości metody (88%) jest spowodowana faktem, iż w trakcie testów część gestów mogła zostać wykonana zbyt wolno lub w zbyt dalekiej odległości od sensora. Kiedy w polu widzenia czujnika obecna jest dłoń wykonująca gest jego pobór mocy sięga 20 mW zaś w czasie oczekiwania na gest wielkość ta spada do 0,3 mW.

Czujnik gestów ruchu (ang. *Motion Gesture Sensor*, MGS) o innej budowie, lecz także przeznaczony do rozpoznawania kierunku przesunięcia dłoni w jednej osi zaproponowali Kim i in. [51]. Motywacją zespołu do podjęcia prac nad nową konfiguracją sprzętową i przestrzenną sensora było zmniejszenie wymiarów fizycznych optycznego czujnika gestów i jego poboru mocy. Zaproponowaną przez nich konstrukcję stanowią, w odróżnieniu od typu czujnika opisanego powyżej, tylko jedno źródła światła w postaci podczerwonej diody LED oraz dwie, położone blisko siebie fotodiody, oddzielone jednak blokiem optycznym (Rys. 2.1b). Z założenia, emitowana przez diodę LED (która umiejscowiona jest poza osią przechodzącą przez fotodiody i blok optyczny, na Rys. 2.1b dioda LED umiejscowiona jest przed blokiem optycznym) prostokątna fala światła odbijana jest przez dłoń wykonującą ruch i, w danym momencie, odbierana tylko

przez jedną z fotodiod lub poziom odbieranego sygnału przez każdą z nich jest zróżnicowany (dopóki dłoń nie znajduje się centralnie nad sensorem). Kierunek ruchu wzdłuż osi sensora ustalany jest na podstawie różnicy czasów odebrania sygnałów przekraczających wartość progową natężenia światła przez fotodiody (ang. *timing difference*, TD). Rolą bloku optycznego jest ograniczenie pola widzenia fotodiod od strony wewnętrznej sensora, dzięki czemu mogą one być umiejscowione bardzo blisko siebie. Zmniejszy się wówczas wielkość tak zwanej *strefy niepewnej* - strefy widzianej przez obydwie fotodiody a zwiększą się *strefy detekcji*, czyli strefy widziane na raz tylko przez jedną z fotodiod. Kiedy obiekt odbijający światło znajduje się w *strefie niepewnej*, sygnały na fotodiodach osiągają wartości progowe równocześnie (TD=0). Brak bloku optycznego powodowałby konieczność rozmieszczenia fotodiod daleko od siebie celem zmniejszenia *strefy niepewnej*. Zastosowanie bloku optycznego może zmniejszyć jednocześnie wielkość *strefy martwej* a więc obszaru niewidocznego dla obydwu czujników. Opisane strefy zostały oznaczone na Rys. 2.1b.



Rys. 2.1 a) Schemat czujnika z dwoma diodami LED i jedną fotodiodą używany przez Chenga [32] (opr. własne).
 b) Schemat czujnika z jedną diodą LED, dwoma fotodiodami i blokiem optycznym Kima [51] (opr. własne).

Odległość l na wysokości h ponad jedną z fotodiod oznacza jej zasięg widzenia czyli jest to długość stanowiąca "przekrój" *strefy detekcji*. Biorąc pod uwagę, iż odległość między fotodiodami oraz wysokość bloku optycznego są pomijalne względem dystansów l i h , długość l można oszacować ze wzoru:

$$l \approx h \cdot \left(\tan \left(\frac{\beta}{2} \right) - \tan \gamma \right), \quad (2.1)$$

gdzie β i γ to kąty widzenia elementu optoelektronicznego ograniczone odpowiednio przez obudowę czujnika oraz blok optyczny. Ze wzoru (2.1) i ze schematu (Rys. 2.1b), co zostało poparte wynikami symulacji i pomiarów zespołu, wynika, iż zmniejszanie *strefy niepewnej*

(czyli zmniejszanie kąta γ poprzez zwiększanie wysokości bloku optycznego) wydłuża *strefy detekcji*, zwiększa efektywny obszar roboczy sensora i dokładność wykrywania ruchów. W trakcie wykonywania eksperymentów założonym celem było znalezienie częstotliwości próbkowania (i zarazem włączania diody LED), dla której w danej konfiguracji (wymiarów i kątów) sensora będzie on wyróżniał się dokładnością rozpoznawania ruchów przesunięcia dłoni na poziomie co najmniej 99,5%. Rezultaty dowiodły, że coraz wyższa częstotliwość próbkowania wymagana jest dla pomiarów wykonywanych na obiektach wykonujących ruchy dalej od sensora, gdyż są one postrzegane przez czujnik jako mniejsze. Jednak dla obiektu o dowolnej szerokości istnieje krytyczna odległość od czujnika kiedy nawet zwiększona częstotliwość próbkowania nie będzie w stanie zrekompensować małej ilości światła odbitego przez obiekt i detekcja ruchu w takim przypadku będzie niemożliwa. W trakcie przeprowadzonych eksperymentów badano częstotliwości pracy sensora w zakresie do 2kHz, oraz o współczynniku wypełnienia diody $D=0,5$ i szczytowym poborze prądu na poziomie 25 mA co przełożyło się na średni pobór prądu o wielkości 12,5 mA. Przy przyjętej mocy diody LED i częstotliwości granicznej czujnika, nie udało się wykryć obiektów o szerokości 20 mm (imitujących palec) w odległościach większych niż 75 mm od sensora, co powiodło się jednak dla obiektu szerokiego na 80 mm (imitującego dłoń). Parametry geometryczne czujnika podczas eksperymentu wynosiły odpowiednio: $L_d=2,4$ mm (odległość pomiędzy środkami fotodiod), $h_{ob}=5,3$ mm (wysokość bloku optycznego), $L_{ob}=1$ mm (szerokość bloku optycznego) oraz $\gamma=7,5^\circ$.

W kolejnej pracy tego zespołu na temat czujnika ruchu MGS autorzy powtórzyli eksperymenty dokładności detekcji ruchu w funkcji szerokości obiektu, jego odległości od sensora i częstotliwości próbkowania [52]. Zmniejszono wymiary bloku optycznego do $L_{ob}=0,4$ mm i $h_{ob}=1,5$ mm aby jego obudowa lepiej nadawała się do cienkich urządzeń przenośnych. Nie spowodowało to istotnej zmiany wartości parametru γ , która pozostała na poziomie $7,5\pm 0,1^\circ$. Zwiększono jednak także zakres analizowanych częstotliwości i wydajność diody LED. W efekcie dla próbkowania na poziomie 25kHz osiągnięto docelową dokładność 99,5% dla wszystkich analizowanych konfiguracji pomiarowych, czyli obiektów o szerokościach 20 i 80 mm umieszczonych w zakresie 50 - 100 mm od sensora. Współczynnik wypełnienia cyklu diody na poziomie $D=0,125$ i szczytowy pobór prądu na poziomie 100 mA przełożyły się na średni pobór prądu przez sensor w granicach 13 mA.

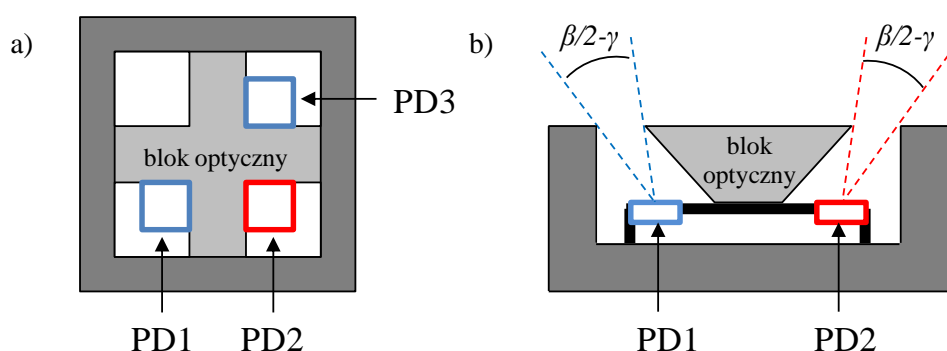
² Wartości nie podano w tekście - obliczenia własne autora na podstawie danych zawartych w artykule.

b) Czujniki kilkudetektorowe

W niniejszej pracy, za czujniki kilkudetektorowe uważane są wszystkie o liczbie detektorów większej niż 2 oraz takie, które w swojej konstrukcji składają się z co najwyżej 10 detektorów lub nie stanowią matrycy większej niż 4x4 elementy światłoczułe.

Rozwój sensora typu MGS zespołu Kim i in. w kierunku uwzględnienia ruchów w następnych dwóch osiach oraz dalszego zmniejszania wysokości bloku optycznego bez pogarszania dokładności detekcji ruchów przez czujnik został opisany w [53]. Detekcja ruchów w kolejnej osi w płaszczyźnie MGS stała się możliwa poprzez dołożenie do jednowymiarowej konfiguracji czujnika dodatkowej fotodiody, tak aby trzy elementy światłoczułe były ułożone względem siebie w kształcie litery „L” (Rys. 2.3a). Dodatkowy element, wraz z jedną z fotodiod z konfiguracji jednowymiarowej MGS utworzy parę rejestratorów nasłuchujących ruchów w drugiej osi w płaszczyźnie czujnika gestów. Jedna z trzech fotodiod układu będzie zatem zaangażowana w pomiary ruchów w obydwu osiach. Detekcja kierunków w osiach na płaszczyźnie odbywa się poprzez analizę sygnału różnicowego sygnałów natężenia światła z fotodiod znajdujących się na rozpatrywanej osi. Kierunek ruchu w osi od/do sensora ustalany jest na podstawie zmienności amplitudy, która obliczana jest jako średnia wartość natężenia na wszystkich trzech fotodiodach. Diagnoza takiego ruchu jest również możliwa w konfiguracji z dwiema fotodiodami, o czym autorzy pisali we wcześniejszych publikacjach. Wciąż jednak do podświetlenia obiektu wystarcza tylko jedna dioda LED. Z kolei dalsze zmniejszanie wymiarów układu scalonego niesie ze sobą konieczność uwzględnienia, iż niższa wysokość bloku optycznego wymusza przysunięcie fotodiod w kierunku bliżej bloku aby zachować wartość parametru γ na pożądanym poziomie. Jednak wymiary fizyczne fotodiod w krytycznym momencie ograniczą możliwość przysunięcia tego elementu bliżej i jedynym rozwiązaniem przy klasycznej konstrukcji bloku będzie zastosowanie mniejszej fotodiody. Spadek powierzchni aktywnej fotodiody powoduje z kolei spadek stosunku sygnału do szumu SNR (ang. *Signal to Noise Ratio*) i pogorszenie parametrów całego sensora. Rozwiązaniem konstrukcyjnym zaproponowanym przez zespół było przeprojektowanie bloków optycznych aby ich kształt był trapezoidalny, skierowany krótszą podstawą do dołu (Rys. 2.2b). Dzięki temu niższy blok optyczny może ograniczać pole widzenia fotodiod i parametr γ na w pełni kontrolowalnym poziomie. Szereg symulacji popartych eksperymentami pozwoliły na wybranie optymalnych parametrów trapezoidalnego bloku

optycznego w nowej konstrukcji w postaci jego wysokości, $h_{ob}=0,4$ mm, długości jego górnej podstawy, $l_T=0,4$ mm, która była równa odległości pomiędzy fotodiodami l_S , co ustaliło wartość kąta γ na $12,7^\circ$. Zmiana ta pozwoliła m.in. na skrócenie minimalnej drogi jaką obiekt musi przebyć aby mógł zostać wykryty ruch, w porównaniu z konfiguracją z prostopadłościennym blokiem optycznym, z 30 na 20 mm. Zachowane zostały dokładność rozpoznawania ruchów na poziomie 99,5%, podobna częstotliwości próbkowania układu i parametry prądowe diody LED. Eksperymenty zasięgu czujnika pokazały, iż nasycę się on kiedy obiekt odbijający światło (biały pasek o szerokości 2 cm) znajduje się w odległościach bliższych niż 3 cm.



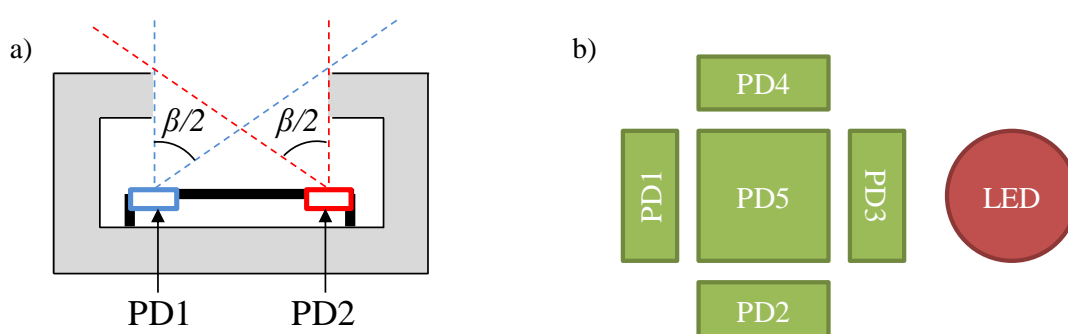
Rys. 2.3 Konfiguracja sensora Kima z trzema fotodiodami [53] (opr. własne). a) Widok z góry. b) Widok z boku.

W swojej kolejnej pracy zespół zaproponował jeszcze inną konstrukcję czujnika ruchów o wydatnie niższym poborze prądu [54]. Celem rekompensaty zmniejszenia prądu sterowania diody LED, rozważono zwiększenie powierzchni fotodiod i rezygnację z bloków optycznych umiejscowionych w środku układu. Przychodząca wiązka światła odbitego była tym razem ograniczana przez obudowę projektowanego elementu. Nowa konstrukcja zawierała 4 podłużne fotodiody (dwie pary, po jednej na oś) ułożone na krawędziach prostokąta. Obudowa elementu, przez którą przez kwadratowy otwór wpada światło, posiada nawisy, zachodzące na połowę szerokości każdej z fotodiod, rozpatrując przekrój elementu wzdłuż którejś z jego osi (Rys. 2.4a). Nawisy ograniczają kąt γ do 0° , zaś odległość między fotodiodami danej pary stanowi teoretycznie szerokość *strefy niepewnej*, która pozostaje stała w funkcji odległości od sensora. Dzięki zwiększeniu powierzchni fotodiod (wydłużenie) i otwarciu przestrzeni elementu (pozbycie się klasycznego bloku optycznego), możliwe stało się zredukowanie mocy nadajnika światła, przy zachowaniu wymaganego poziomu sygnału na fotorejestratorach. Autorzy podają, iż całkowity pobór prądu przez sensor został

zmniejszony do zaledwie 3,78 mA co jednocześnie pozwoliło na odnotowanie dokładności detekcji ruchów w odległości do 10 cm od sensora na poziomie 100%.

Literatura prezentuje również inne, zróżnicowane co do zasady działania i możliwości, rozwiązania kilkudetektorowych optycznych czujników gestów, spośród których część przedstawiono poniżej.

Konstrukcję odbiciowego czujnika gestów złożoną z pięciu fotodiod (PD1-PD5) i jednej diody LED zaproponował Zivkovic [55]. Cztery prostokątne elementy światłoczułe ułożone są wzdłuż czterech ścianek kwadratowej środkowej fotodiody. Pulsacyjne źródło światła umieszczone jest bezpośrednio obok detektorów, we wspólnej obudowie (Rys. 2.4b).



Rys. 2.4 a) Sensor ruchu bez bloków optycznych Kima [54] (opracowanie własne).
b) Budowa sensora Zivkovica [55] (opracowanie własne).

W fizycznej implementacji panele odbiorcze czujnika są jeszcze otoczone ścianką, która stanowi część obudowy całego sensora. Dzięki jej obecności ruchy wykonywane przez dłoń w danej osi sensora generują sygnały o nieco odmiennych kształtach na poszczególnych fotodiodach w osi. Ścianka zaciemnia bowiem w różnym stopniu poszczególne panele odbiorcze przed odbijanym światłem podczerwonym. Dzięki temu możliwe jest określenie kierunku ruchu, mimo bardzo bliskiego wzajemnego położenia elementów. Podczas badań testowano również wpływ nałożenia na czujnik układu optycznego w postaci otworu (*pinhole*), którego zadaniem było ogniskowanie odbitego światła na panelach czujnika celem uwypuklenia różnic w stopniu ich oświetlenia. W trakcie przeprowadzonych eksperymentów zbadano dokładność sensora w rozróżnianiu ruchów dłoni w dwóch osiach w płaszczyźnie czujnika (lewo / prawo i góra / dół) oraz w detekcji fazy zbliżania dłoni do sensora. Dla każdego z tych gestów zarejestrowanych zostało po 600 nagrań o różnych prędkościach ruchu i przy różnych ułożeniach dłoni wykonującej gest. Uzyskana baza gestów została podzielona po równo na zbiór uczący i testowy. Dla pełnego zestawu pięciu gestów i konfiguracji czujnika z dodatkową optyką

osiągnięto ponad 99,3% dokładności rozpoznawania gestów. W wersji pozbawionej optyki dokładność ta wynosiła 93% co może być spadkiem akceptowalnym w porównaniu z nakładami potrzebnymi do wzbogacenia czujnik o dodatkową optykę. Ponadto Zivkovic zauważa, iż zaproponowany sensor gestów, kiedy byłby użyty przy urządzeniu posiadającym własny wyświetlacz, mógłby być pozbawiony swojego źródła światła. Jako podświetlenie służyłoby wówczas modulowane światło pochodzące z wyświetlacza [56].

Chuang i in. zamontowali pary nadajników i odbiorników światła IR w czterech narożnikach urządzenia mobilnego i dzięki wykorzystaniu trilateracji i filtra Kalmana byli w stanie precyzyjnie lokalizować pozycję ręki względem ekranu urządzenia [57][58]. Odchylenie standardowe pozycji nie przekraczało 0,03 cm dla żadnej z trzech rozpatrywanych osi. W publikacji nie rozważano użycia układu do rozpoznawania gestów dyskretnych.

Interfejs do multidotykowej interakcji z urządzeniem mobilnym, montowany na krawędziach jego obudowy zaproponował Butler i in. [59]. Wykonali oni dwie konstrukcje liniowo ułożonych 10 odbiciowych czujników Avago HSDL 9100-021 [42] i umieścili po jednej na każdej z podłużnych krawędzi urządzenia mobilnego. Palce pojawiające się w pobliżu linijki czujników odbijają światło do każdego sensora w różnym stopniu, w zależności od wzajemnego położenia (kąta i odległości) generując na każdym z nich odpowiednią wartość stowarzyszoną z intensywnością odbieranego światła. W efekcie, powstaje przekrój, obraz 1D, na podstawie którego autorzy obliczają środek ciężkości zarejestrowanego profilu i szacują położenie palca / dłoni względem czujnika w osi równoległej do krawędzi. Dodatkowo, autorzy proponują gesty dwuręczne, do wykonania z obydwu stron urządzenia mobilnego na raz: gesty zbliżania, przesuwania i rotacji. Są to typowe gesty ciągłe, zaś z gestów dyskretnych wspomniano jedynie o kliknięciu. W pracy nie opisano modelu odpowiedzialnego za rozpoznawanie gestów oraz dokładności ich detekcji.

Podobne rozwiązanie, liniowy układ 10 par nadajników i odbiorników światła podczerwonego (samodzielnej konstrukcji) zaproponowali Tang i in. [60]. Interfejs w rozważanym przypadku użycia montowany był na bocznej krawędzi komputera przenośnego tak, aby w jego polu widzenia znajdowała się przestrzeń, w której najczęściej operuje się myszą komputerową. Czujnik zaprojektowano do śledzenia ruchów ręki i obsługi trzech rodzajów wirtualnych kliknięć. Wykorzystanie modelu skończonej maszyny stanów

(FSM) pozwoliło na uzyskanie dokładności klasyfikacji gestów kliknięcia, podwójnego kliknięcia i przeciągnięcia (ang. *drag*) na poziomie odpowiednio 84%, 70% i 72%.

Analogiczna konstrukcja, 8 par liniowo ułożonych nadajników i odbiorników światła, wykorzystana została również do innego celu. Skanowanie konturów narysowanych na płaszczyźnie oraz rejestrację kształtów w 3D opisali Tar i Cserey [61]. Dwa segmenty po 3 pary nadajników i odbiorników światła podczerwonego, zamieszczone naprzeciw siebie wykorzystano do obliczania parametrów walca (np. prędkość, wielkość, współczynnik odbicia światła) przemieszczającego się między nimi [62].

Czujnik z trzema emiterami światła rozmieszczonymi równomiernie na okręgu o promieniu 2 cm i jednym odbiornikiem rejestrującym trzy kanały (odbicie światła pochodzące od każdego z nadajników z osobna) umieszczonym w jego środku opisali Batchuluun i in. [63]. Powstałe podczas wykonywania gestów profile zostały sparametryzowane i podane jako wejście do systemu ANFIS (ang. *Adaptive Network-based Fuzzy Inference System*), który integruje właściwości sieci neuronowych i logiki rozmytej. Zaprezentowane zostały wyniki dokładności detekcji 4 gestów dyskretnych: przesunięcie dłoni, uderzenie, rotacja i łaskotanie, która po zaokrągleniu części tysięcznej wyniosła 100%. Podobny sensor, lecz o trzech parach LED - fotodiody, umieszczonych na okręgu co 120° wykorzystano do rozpoznawania kierunku wykonywania gestów rotacji dłonią [64].

Zestaw czujników w postaci par fotodiod i diod LED IR lecz w różnych konfiguracjach geometrycznych, nazwany jako zSense, zaproponowali Withana i in. [65]. Rozważyli oni rozłożenie par przetworników na wierzchołkach trójkąta prostokątnego, w linii i na pierścieniu (jako interfejs zakładany na palec). W odróżnieniu od wielu metod, zSense nie zapala wszystkich diod LED na raz, lecz w różnych konfiguracjach, tworząc dzięki temu zróżnicowane profile podświetlenia dla tej samej pozy dłoni. Celem klasyfikacji otrzymanych profili zastosowano dwuetapowy proces wykorzystując metodę maszyny wektorów nośnych oraz sieci Bayesowskie. Rozważono w sumie rozpoznawanie 11 gestów dyskretnych, w tym także statyczne (dłoń w konfiguracjach: pojedynczy palec, dwa palce rozpostarte, trzy palce rozpostarte; niech konfiguracje te będą nazywane odpowiednio: 1FS, 2FS, 3FS), co jest rzadkością w przypadku kilkuelementowych czujników gestów. W finalnej wersji systemu, w zależności od konfiguracji geometrycznej, rozpoznawano do 7 gestów dyskretnych ze dokładnością 94,8%, uśredniając wyniki dla

wszystkich trzech rodzajów czujnika. zSense wysyłał dane do komputera PC z częstotliwością 50Hz a zapotrzebowanie energetyczne jego elementów optoelektronicznych (nie wliczono zużycia energii przez mikroprocesor) wynosiło od 7,2 mW do 29 mW w zależności od konfiguracji.

W ramach rozwoju badań nad bezdotykową interakcją Withana i in. zaproponowali modyfikację zSense, o nazwie waveSense, jako bezdotykowy interfejs do gogli do wirtualnej rzeczywistości (ang. *Virtual Reality*, VR) [66]. Pary przetworników umieszczone zostały w specjalnej ramce okalającej gogle VR. System potrafił lokalizować dłoń umieszczoną w przestrzeni przed czujnikiem, która została podzielona na 40 (5·4·2, w osiach x, y, z) woluminów z dokładnością przypisania ok. 95%.

Aktywny czujnik gestów w postaci peryferyjnego rozszerzenia do urządzeń wyposażonych w system Android zaproponował Zhang i in. [67]. Ich czujnik składa się z liniowo ułożonych elementów optoelektronicznych: dwóch fotodiod i jednej diody LED umieszczonej pomiędzy czujnikami światła. Za pomocą systemu wykrywano kliknięcia palcem na wirtualnej klawiaturze z dokładnością ponad 90% (błąd lokalizacji był mniejszy niż 1 cm). Czujnik pozwolił również na rozpoznawanie znaków (liter) rysowanych palcem na wirtualnym trackpadzie z dokładnością 90,6%.

Część rozwiązań dotyczących optycznych czujników wielopunktowych również, jak układy opisane w poprzedniej sekcji, zostało wprowadzonych do masowej produkcji. Są one obecnie powszechnie wykorzystywane choćby w multimedialnych telefonach i zegarkach, tabletach ale także w autonomicznych robotach. Jednym z takich przykładów jest element Sharp GP2AP054A00F, składający się z czterech fotodiod ułożonych w kształt krzyża oraz jednej diody LED [68]. Kolejnym rozwiązaniem jest czujnik APDS 9960 [69], posiadający 4 fotodiody do detekcji kierunków ruchu oraz także 4 fotodiody do wykrywania różnych kolorów i jedną diodę LED. Innym przykładem jest sensor fotometryczny ADUX1020, zdolny do określania kąta padania wiązki światła, wykorzystywany również do interakcji za pomocą gestów ciągłych [70].

Czujnik gestów montowany na zegarku wykorzystujący 4 pary podczerwonych zbliżeniowych sensorów Sharp GP2Y0D340K zamontowanych na krawędziach kwadratowej ramy opisali Kim i in. [71]. Jako model analizy gestów zastosowano narzędzie bazujące na modelu HMM. Dla czterech różnych warunków oświetlenia uzyskano średnią dokładność rozpoznawania 5 gestów na poziomie 95,5%.

Interakcja naokoło urządzenia (telefonu multimedialnego lub zegarka) za pomocą aktywnych czujników zbliżeniowych Sharp GP2D120X opisali Kratz i Rohs [72]. W zaproponowanej konfiguracji zamieszczono po 3 przetworniki na każdej z podłużnych krawędzi smartfona. Wykorzystano metodę DTW (ang. *Dynamic Time Warping*) do klasyfikacji gestu względem wzorcowych przebiegów. Dla 9 gestów uzyskano dokładność klasyfikacji na poziomie 88,6%.

Czujnik matrycowy o wielkości 3x3 złożony z fotodiod i pracujący w trybie pasywnym został zaprojektowany przez Kaholokula [73]. Wykorzystany został klasyfikator k-NN (ang. *k-Nearest Neighbors*, pl. *K-Najbliższych Sąsiadów*), za pomocą którego układ rozpoznawał 10 dyskretnych dynamicznych gestów ze dokładnością 98%. Czujnik pracuje skutecznie w szerokim zakresie intensywności światła widzialnego lecz nie ma możliwości rozpoznawać gestów podczas ciemności oraz w bardzo jasnych światłach.

Nieco innego rodzaju optycznym sensorem jest czujnik PIR (ang. *passive infrared*), który jest elektronicznym czujnikiem mierzącym światło podczerwone (IR) promieniujące z obiektów znajdujących się w jego polu widzenia. Zasada działania opiera się na tym, że każdy rzeczywisty obiekt emituje energię cieplną właśnie w postaci promieniowania w zakresie podczerwonym. Matrycę czujników PIR o wielkości 4x4 w zastosowaniu do wykrywania ruchów dłoni rozważał Wojtczuk i in. [74]. Wykrywali oni w odległości do 20 cm od urządzenia 4 dynamiczne gesty dłoni z dokładnością 92,6%. W ograniczonej do matrycy 2x2 wersji czujnika uczeni zastosowali także ograniczenia w polu widzenia poszczególnych czujników PIR pozwalające na poprawę detekcji kierunku ruchu. Badania wykazały 77% poprawnej klasyfikacji wykonanych gestów oraz 86% poprawnej detekcji kierunku wykonanego przesunięcia dłonią [75].

c) Czujniki wielodetektorowe

Metody rozpoznawania gestów, bez względu na wymagania dotyczące zasobów obliczeniowych czy energetycznych, od początków badań interakcji człowieka z komputerem opierały się o zastosowanie sygnału video z kamer. Rozwiązania pozwalające na efektywne algorytmicznie znajdowanie zarysu dłoni w obrębie danej klatki sygnału wideo często bazują na wykonywaniu gestów przez dłoń ubraną w kolorową rękawiczkę [76][77][78]. W takim wypadku znalezienie zarysu dłoni w obrazie dokonuje się przez detekcję koloru rękawiczki, najczęściej jaskrawego, o małym prawdopodobieństwie wystąpienia na scenie w postaci innych przedmiotów. Jest to jednak metoda wymagająca kontaktu dłoni

z materiałem rękawiczki, więc może być niewygodna w użyciu i nie nadająca się z tych względów do wielu zastosowań. Rozwiązania systemów rozpoznawania gestów bazujące na analizie obrazu nie ubranej dłoni cechują się zaletą bezkontaktowości lecz wymagają użycia nieco innych algorytmów ekstrakcji sylwetki dłoni i są bardziej podatne na zmiany oświetlenia i różnorodność sceny [79][80][81][82]. System wielowidokowej kamery został wykorzystany przez Ueda i in. w celu stworzenia wokselowego modelu dłoni i dopasowania go do modelu szkieletu [83]. Rozpoznawanie gestów na podstawie analizy cienia dłoni z wykorzystaniem kamery RGB wyposażonej w filtr IR przeprowadził Rekimoto [84]. Czujnik gestów o nazwie Leap Motion składający się z dwóch kamer obserwujących dłońe jednorodnie podświetlone przez podczerwone diody LED wykorzystywany był zaś w literaturze między innymi do rozpoznawania gestów niedokończonych oraz znaków języka migowego [85][86].

Wraz z rozwojem technologii na rynku urządzeń elektronicznych pojawiły się kamery głębi, w obrazie z których intensywność piksela tworzącego dany obiekt sceny jest proporcjonalna do odległości tego obiektu od obiektywu przetwornika. Zastosowanie progowania obrazu głębi ze względu na poziom jasności pikseli pozwalana na efektywne obliczeniowo znajdowanie obiektów położonych bliżej kamery, przykładowo wyciągniętych w jej kierunku dłoni, na tle sceny. Jedną z najbardziej powszechnie stosowanych w badaniach naukowych kamer głębi jest Microsoft Kinect. Jego pierwsza generacja bazuje na analizie podczerwonego światła strukturalnego (kamera mierząca deformację wysświetlanej chmury punktów przez obecność obiektów w przestrzeni) zaś druga zawiera przetwornik ToF (ang. *Time of Flight*) (modulacja wiązki z nośnikiem częstotliwości radiowej i pomiar przesunięcia fazowego nośnika po stronie odbiornika). Kamery Kinect wykorzystywano między innymi do rozpoznawania statycznych póz [87], języka migowego [88][89] czy obsługi robotów [90]. W badaniach Gupty i in. do rozpoznawania dłoni wykorzystywano dane z kamery głębi SoftKinetic [91]. Prosty czujnik głębi oparty o układ jednej diody LED, soczewki i matrycy CCD do śledzenia ruchów palca lub dłoni opisali Ryu i in. [92].

Dzięki powstaniu wydajniejszych procesorów i minimalizacji kamer (także głębi) rozpoznawanie gestów oparte na analizie wideo zostało również wdrożone na platformach mobilnych. Lu i in. rozpoznawali interakcje używając smartfona do detekcji dwóch połączonych palców na obrazie wideo [93]. Rozpoznawanie dziesięciu symboli

pokazywanych przez dłoń na telefonie z platformą Android pokazali Lahiani i in. [94]. Smartfon z kamerą z dwoma obiektywami i transformacją odległości rozpoznający głębię sceny zostały wykorzystane przez Wong i in., którzy analizowali gesty oparte na liczbie i lokalizacji aktywnych opuszków palców ręki [95].

2.1.2. Czujniki gestów inne niż optyczne

Przywołując podział czujników według Berman [34], w tej sekcji wyróżniono przykłady bezdotykowych czujników gestów bazujących na pobudzeniach innego rodzaju niż optycznych. Ze względu na cechę bezdotykowości nie wspomniano o czujnikach mechanicznych. Zaprezentowane zostały zatem przykłady czujników pojemnościowych (elektryczne), elektromagnetycznych i akustycznych.

Pojemnościowe czujniki gestów

Jeden ze sposobów śledzenia ruchów ręki lub palca w przestrzeni trójwymiarowej wykorzystuje analizę zmian pojemności między parami elektrod, wywołanymi przez przemieszczające się w ich pobliżu ożywione tkanki jak np. dłoń. Rozwiązanie zaprezentowane zostało m.in. przez Aezinia [96]. Technika różnicowego pojemnościowego pomiaru poprzez ułożenie trzech elektrod w linii pozwala na śledzenie obiektu w jednej osi, wzdłuż sensora. Ponadto, taka konfiguracja jest również w stanie wykryć prostopadły do sensora kierunek ruchu. Jej rozdzielczość pozwala na rozróżnienie czy gest wykonywany był jednym palcem czy całą dłonią, lecz nie jest wystarczająca (głównie ze względu na ilość zastosowanych elektrod) do rozpoznania konfiguracji palców dłoni (jak wiele ich jest, czy są złączone).

Dwuosiowy sensor gestów bazujący na pojemnościowo kontrolowanych oscylatorach pierścieniowych został zaprezentowany przez Pulkkinena i in. [97]. Przetwornik tego czujnika składa się z trzech elektrod a cały układ zużywa zaledwie 462 nW przy zasilaniu napięciem na poziomie 0,8 V. Układ był testowany dla gestów typu przesunięcie dłoni w prawo i w lewo dla trzech różnych dystansów między dłonią a sensorem (~1,5 cm, ~4 cm, ~6 cm). Wszystkie 108 zdarzeń wykonanych przez ochotników zostało zaklasyfikowanych poprawnie.

Elektromagnetyczne czujniki gestów

System rozpoznawania gestów AllSee wyodrębnia informacje o gestach z interferencji ręki z otaczającymi sygnałami bezprzewodowymi, np. RFID [98]. Może on wykryć 8 rodzajów dynamicznych gestów, które są głównie oparte na zmianach bliskości dłoni, ale czujnik jest również w stanie wykryć takie akcje, jak pstryknięcie palcami.

Grupa Zaawansowanych Technologii i Projektów firmy Google opracowała czujnik gestów, który działa jako radar zniekształceń fal radiowych o wysokiej częstotliwości spowodowanych interakcją palców, znany pod nazwą Project Soli [99]. Wśród innych ruchów czujnik może wykryć specjalny gest, jak pocieranie palcami, co jest trudne do osiągnięcia dzięki innej technologii wykrywania.

Czujnik bazujący na wygenerowanych i odbitych od dłoni wykonującej gest sygnałach impulsów radiowych, wykorzystujący sieci neuronowe i rozpoznający 6 statycznych gestów ze dokładnością 91% zaproponowali Kim i in. [100].

Czujniki gestów dłoni wykorzystujące radar dopplerowski fali ciągłej oraz nadajniki WiFi badał zespół Fana i in. [101][102]. Tang i in. wykorzystali retransmisję przechwyconych sygnałów WiFi i sygnał echa odbijanego od ręki [103]. Ich system wykorzystujący pasywną technologię radarową i polegający na efekcie Dopplera wykrywa pięć dynamicznych dyskretnych gestów. System został następnie rozszerzony o wykorzystanie sygnałów LTE [104]. Rozpoznawanie gestów kołowych i półkolistych w trzech wymiarach za pomocą bezkontaktowego czujnika pasywnych znaczników RFID (ang. *Radio Frequency Identification*) zostało zbadane przez Krigslunda i in. [105].

Akustyczne czujniki gestów

Czujnik do śledzenia lokalizacji dłoni wykorzystujący fale ultradźwiękowe generowane przez smartfon został rozważony przez Dama i in. [106]. System umożliwia lokalizację przestrzenną dłoni w jednym z 8 woluminów ($2 \cdot 2 \cdot 2$, w osiach x , y , z) z dokładnością 77%. Zastosowanie drzewa decyzyjnego bazującego na wyodrębnionych cechach sygnałów pozwoliło na rozpoznanie 8 gestów z dokładnością między 87 a 100%.

Badanie możliwości wykorzystania czujników ultradźwiękowych jako radarów obiektów i systemu nasłuchiwanie gestów zostało również przeprowadzone w porównawczym zestawieniu z aktywnym optycznym czujnikiem gestów o konstrukcji liniowej [107].

2.1.3. Podsumowanie

Przy projektowaniu optycznego czujnika gestów należy zwrócić uwagę na szereg parametrów, za pomocą których można dokładnie opisać jego cechy. Wyróżniono trzy podstawowe grupy parametrów:

- geometryczno-przestrzenne parametry czujnika,
- elektryczne właściwości czujnika,
- funkcjonalne możliwości czujnika.

Do grupy pierwszej zaliczyć można wymiary fizyczne, pole widzenia, strefy operacyjne, zasięg czy rozdzielczość przestrzenną. Jednym z głównych parametrów jest *strefa detekcji* sensora, wyrażona kątowno lub w formie funkcji szerokości strefy od odległości od czujnika. W przypadku czujników aktywnych istotne będzie także określenie *strefy martwej* iluminatora, w której to kiedy obiekt się znajdzie, nie zostanie oświetlony a w efekcie jego widoczność dla czujnika będzie zerowa.

W kontekście parametrów grupy drugiej należy zwrócić uwagę na częstotliwość próbkowania czujnika, poziom napięcia zasilania, rozdzielczość przetwornika analogowo-cyfrowego, pobór prądu przez poszczególne elementy i możliwości jego ograniczenia.

Grupa trzecia zawiera zaś liczbę gestów dyskretnych obsługiwanych przez dany czujnik oraz opis jego potencjalnych możliwości wykorzystania jako interfejsu gestów ciągłych (np. zdolność do śledzenia ruchu dłoni w osi x i błąd oszacowania położenia).

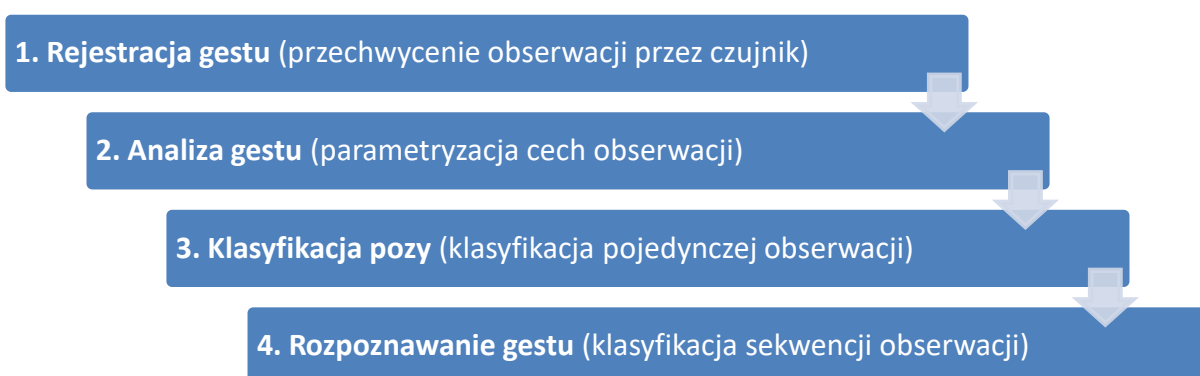
2.2. Rozpoznawanie gestu jako proces

Poglądowe przedstawienie typów czujników gestów obecnych w literaturze należy uzupełnić o opis możliwych etapów toru przetwarzania sygnałów przez nie rejestrowanych.

Obok taksonomii, innym aspektem porządkowania wiedzy na temat gestów jest analiza ruchów, które je tworzą oraz ich interpretacja. W nauce bada się jak człowiek poprzez gestykulację czy mimikę przekazuje informacje. Dziedzinę nauki, która zajmuje się komunikacją niewerbalną pomiędzy ludźmi jest kinezyka. Człowiek patrzący w kierunku osoby wykonującej gest mimowolnie rejestruje go za pomocą receptorów wzrokowych. W zależności od luminancji postrzeganej przez oczy, źrenica oka dostosowuje swoją średnicę co jest elementem adaptacji systemu i przetwarzania wstępnego. Odbierany sygnał z czopków i pręcików jest następnie przetwarzany na sygnał elektryczny i przesyłany do mózgu gdzie następuje interpretacja sekwencji kolejnych, często zróżnicowanych względem siebie, faz gestu przez porównanie z nauczonymi wcześniej wzorcami. W podobny sposób

można rozpatrywać etapy rozpoznawania gestów przez elektroniczny sensor gestów, niezależnie od charakteru wielkości fizycznej przez niego rejestrowanej.

W wielofunkcyjnych urządzeniach mobilnych wyróżnić można sprzętowe komponenty lub programowe wątki odpowiedzialne, co najmniej częściowo, za rozpoznanie gestu. Sekcję taką można nazwać *systemem rozpoznawania gestu* (ang. *Gesture Recognition System*, GRS). Jako jedni z pierwszych pojęcie to wykorzystali Voyles i Khoslas opisując część urządzenia do owijania przewodów [24]. Pojęcie GRS przewijało się w literaturze w kontekście konkretnych systemów, najczęściej opartych na analizie wideo [13]. Przy przeglądzie rozwiązań systemów wizyjnych Pavlovic wyróżnił składowe GRS jako: model gestu, analizę danych, rozpoznanie gestu i fazę aplikacyjną [108]. Berman i Stern zaproponowali pojęcie kontrolera GRS, którego opis jest precyzyjny co do modułowości systemu i niezależny od typu sensora [34]. Rozumieją oni kontroler GRS jako sekwencyjną realizację bloków funkcjonalno zdarzeniowych, na które składają się: faza pokazania gestu, przechwycenie gestu przez czujnik, analiza gestu (śledzenie pozycji dla gestów dynamicznych oraz parametryzacja cech), rozpoznanie gestu i maszyna stanów realizująca odpowiednie reakcje systemu. Na potrzeby niniejszej rozprawy, schemat tego systemu został zaadaptowany do ogólnej postaci zawężonej do czterech faz (Rys. 2.5).



Rys. 2.5 Elementy składowe systemu rozpoznawania gestów (GRS) zaadaptowanego ze schematu Berman i in. [34].

Dodatkowo, Berman i Stern zaznaczyli, że GRS zaprezentować można jako system kontrolny, czyli system zarządzania lub sterowania zachowaniami innych systemów lub urządzeń. W takiej postaci uwidacznia się rola sensora gestów jako pośrednika między człowiekiem wykonującym gesty a systemem nasłuchującym i gotowym do reakcji na nie.

Zgodnie z tak skonstruowaną definicją GRS, system powinien zatem zawierać urządzenie rejestrujące gesty jak i mieć dostęp do procesora przetwarzającego dane. Wymagane jest także zaprojektowanie dedykowanej metody klasyfikacji sekwencji oraz modelu uwzględniającego kontekst systemowy. Model taki może być zawarty w obrębie czujnika gestów lub w urządzeniu będącym klientem nadchodzących komunikatów o wykrytych gestach. Kontekst systemowy oznacza przypisywanie tym samym gestom różnych akcji, lub pomijanie części gestów, w zależności od konkretnego widoku czy typu aplikacji obsługiwanej w danym momencie przez czujnik.

2.3. Metody rozpoznawania póż i gestów

Pierwsze dwa etapy GRS dotyczą pokazania gestu i jego rejestracji za pomocą czujnika. W niniejszej pracy, typ czujnika został ustalony (liniowy czujnik optyczny) i zostanie on szczegółowo opisany w Rozdziale 3. Kolejne etapy dotyczą ekstrakcji cech przechwyconych sygnałów. Sposób przetwarzania i ewentualna parametryzacja determinowana będzie wymaganiami co do sygnałów wejściowych przez metody klasyfikacji póż i gestów użyte w kolejnych etapach GRS. Z tego względu w kontekście projektowania systemu rozpoznawania gestów najpierw opisano wybrane modele analizy danych.

W pierwszej części tej sekcji zaprezentowano metody klasyfikacji przeznaczone do danych stacjonarnych (jedna obserwacja), często stosowane w literaturze w zagadnieniach rozpoznawania póż. Druga część sekcji zawiera przegląd metod dedykowanych do klasyfikacji sekwencji w czasie, wykorzystywanych do rozpoznawania całych gestów dyskretnych (wiele obserwacji).

Część z opisanych metod klasyfikacji obserwacji stacjonarnych i sekwencji to modele z grupy metod uczenia maszynowego (ang. *machine learning*). Polegają one na dobraniu wartości zmiennych modelu tak aby modele rozróżniały prezentowane wzorce według klas. Ustalanie wartości zmiennych odbywa się poprzez proces zwany uczeniem. Popularną metodą uczenia wzorców jest uczenie nadzorowane (ang. *supervised learning*), w którym model otrzymuje zestaw: dane wejściowe i oczekiwane wyjście. Wykorzystanie dużego zbioru takich zestawów i zastosowanie adekwatnego do modelu mechanizmu uczenia umożliwia wytrenowanie modelu tak, że będzie on w stanie z wysoką dokładnością zaklasyfikować również nieznanne mu dotąd zestawy danych wejściowych. W zagadnieniach weryfikacji jakości modelu zbiory danych dzieli się na zbiór treningowy, walidacyjny i testowy. Model uczy się na zbiorze treningowym dopóki nie zacznie się przeuczać, co sprawdza się

na zbiorze walidacyjnym. Przeuczenie (lub nasycenie, nadmierne dopasowanie, ang. *overfitting*) następuje kiedy w wyniku zbyt długiego treningu model nadto dokładnie rozróżnia obserwacje poszczególnych klas w zbiorze treningowym. W efekcie dopasowana do zbioru treningowego reguła klasyfikacji będzie klasyfikować przypadki z nieznanego sobie dotąd zbioru (walidacyjnego czy testowego) z wyraźnie niższą dokładnością. W uczeniu maszynowym stosuje się zatem różne metody zwiększające zdolność modelu do dobrego działania na wcześniej nieobserwowanych danych (tzw. metody regularyzacji), czyli do uogólniania (ang. *generalization*), np. odrzucanie (ang. *dropout*) czy powiększanie zbioru danych (ang. *data augmentation*) [109]. Jako dokładność wytrenowanego modelu podaje się zaś wynik klasyfikacji uzyskany na zbiorze testowym.

Bardzo istotnym zagadnieniem przy projektowaniu modeli jest także dobór wartości poszczególnych hiperparametrów je opisujących. Hiperparametrem może być każda nietrenowalna wielkość lub właściwość modelu np. ilość warstw, tempo uczenia, rodzaj funkcji przejścia. Różne topologie modelu mogą bowiem dawać różne wyniki dokładności klasyfikacji tego samego zbioru danych. Istnieją dwie grupy metod zautomatyzowanego poszukiwania topologii dającej najwyższą dokładność klasyfikacji. Są to poszukiwanie siatkowe (ang. *grid search*) oraz poszukiwanie losowe (ang. *random search*). Poszukiwanie siatkowe polega na wybraniu rozważanych hiperparametrów modelu, zdefiniowaniu dla każdego z nich zbioru wartości lub dziedziny (punkt początkowy i końcowy) oraz rozdzielczości kroku i sprawdzenie dokładności modelu wytrenowanego dla każdej możliwej kombinacji wartości tak opisanych hiperparametrów [110]. Zaproponowana kilka lat później przez Bergstrę i Bengio metoda poszukiwania losowego wykazała znajdowanie dokładniejszej topologii przy mniejszej liczbie sprawdzonych zestawów hiperparametrów niż metoda poszukiwania siatkowego [111]. Pozwala ona zatem na bardziej wydajne poszukiwania modelu, gdyż statystycznie znajduje dokładniejszy model w krótszym czasie. Poszukiwanie losowe różni się od siatkowego tym, iż nie wybiera się wartości hiperparametrów po kolei z zadanyim krokiem lecz się je losuje. Uprzednio, dla zdefiniowanej dziedziny bądź zbioru wartości hiperparametru określa się charakter rozkładu prawdopodobieństwa wylosowania wartości (np. jednorodny lub log-normalny).

2.3.1. Metody klasyfikacji jednej obserwacji

Zagadnienie rozpoznawania danej pozy ciała lub dłoni polega na interpretacji pojedynczego zestawu danych reprezentujących pozę w danej chwili czasu. W tym celu dokonano przeglądu wybranych metod w dużej mierze stosowanych do klasyfikacji danych stacjonarnych, czyli jednej obserwacji.

a) *k-Najbliższych Sąsiadów*

Metoda *k-Najbliższych Sąsiadów* (ang. *k-Nearest Neighbors*, *kNN*) jest klasyfikatorem, za pomocą którego możliwe jest skojarzenie danej obserwacji (najczęściej wyrażonej w postaci wektora N_c cech) z jedną z grup klas na podstawie odległości pomiędzy wektorem obserwacji i reprezentantami danej klasy w przestrzeni cech. Parametr k oznacza liczbę pierwszych sąsiadów danej obserwacji, posortowanych rosnąco względem obliczonej odległości od rozważanej obserwacji do reprezentantów klas w N_c wymiarowej przestrzeni. Obserwacja zostaje zaklasyfikowana do klasy, która posiada najwięcej reprezentantów wśród k najbliższych sąsiadów. Liczba k determinuje różne wersje klasyfikatora. Możliwe jest restrykcyjne podejście i analiza tylko najbliższego sąsiada ($k=1$), lecz często używa się arbitralnie liczby $k=5$ [112] lub poszukuje się jej optymalnej wartości [113].

Metodę wykorzystano jako klasyfikator póz dłoni przy danych z kamery RGB-D [112]. Metoda była używana także do rozpoznawania gestów na podstawie sygnałów z kilku fotodiod [73] czy jako etap klasyfikacji gestów, dzięki łączeniu cech trajektorii ruchu i cech pozy dłoni [113]. Metoda *kNN* wymaga zbioru obserwacji różnych klas, do którego elementów potrzebuje się odwołać przy klasyfikacji każdej nowej obserwacji. Jej efektem nie jest wytrenowany zestaw wag, lub ekwiwalent, który mógłby być użyty podczas ewaluacji.

b) *Drzewa decyzyjne*

Metody z rodziny drzew decyzyjnych mają na celu wyodrębnianie wiedzy z podanego zestawu obserwacji opisanego za pomocą wektora cech (atrybutów). Algorytmy drzew decyzyjnych grupują obserwacje według etykiet (klas) maksymalizując dokładność klasyfikacji względem kryterium jednej cechy na każdym etapie. Kryterium nazywane jest węzłem, a każdy węzeł może prowadzić do kolejnych, w których rozpatrywana będzie już tylko część początkowego zbioru, spełniająca kryterium węzła nadrzędnego. W momencie

kiedy nie da się utworzyć już żadnego węzła tworzony jest liść, zawierający etykietę. Jedno drzewo może posiadać wiele liści o tej samej etykiecie i dlatego jest to klasyfikator nieliniowy. Istnieje wiele modyfikacji drzew decyzyjnych jak losowe drzewa (ang. *random tree*) i losowe lasy (ang. *random forest*). Algorytmy te wykorzystywano do rozpoznawania gestów m.in. przez [32][33][99][106].

Metody te można nauczyć wzorców i ich efektem jest zestaw reguł, przez który musi przejść wektor cech nowej obserwacji. Jest zatem bardziej oszczędna obliczeniowo w czasie ewaluacji niż metoda kNN i stanowi jeden z mniej złożonych obliczeniowo przykładów metod uczenia maszynowego. Ponadto, zdolność drzewa decyzyjnego do generalizacji spada wraz ze wzrostem drzewa (głębokością zagnieżdżenia węzłów) z czego można skorzystać jako z hiperparametru podczas procesu uczenia modelu.

c) Dopasowywanie wzorców

W metodzie dopasowywania obserwacji do wzorców (ang. *template matching*) dla każdej z rozpoznawanych klas zdefiniowany jest wzorcowy zestaw cech lub wzorcowa instancja obserwacji, z którymi porównuje się reprezentację nowej obserwacji. W tym celu należy wybrać miarę, dzięki której obliczony zostanie stopień podobieństwa między nową obserwacją a każdą ze wzorcowych. Nowa obserwacja zostanie zaklasyfikowana jako reprezentant klasy, względem której podobieństwo będzie najwyższe. Jako miarę podobieństwa dla obserwacji statycznej (wektor cech, profil) stosuje się np. odległość Machalanobisa. Możliwe jest jednak także porównanie dwóch przebiegów jednego parametru w czasie o różnej długości np. używając DTW (ang. *Dynamic Time Warping*). Metoda dopasowywania wzorca z miarą DTW wykorzystywana była do klasyfikacji gestów między innymi przez [72][114]. Klasyfikacja gestów z miarą współczynnika korelacji i minimum (odległość Manhattańska) zastosowano w [79]. Zaawansowaną modyfikację metody dopasowywania wzorca w rozpoznawaniu pozy dłoni, opartą na detekcji konturu dłoni opisał Lv i in. [93].

Metoda dopasowywania wzorca może być zestawiona z kNN dla $k=1$ oraz kiedy uśredni się wszystkie reprezentacje danej klasy w zbiorze danych tak aby liczba obserwacji wzorcowych odpowiadała dokładnie liczbie klas. Wymagania metody dopasowywania wzorca co do złożoności obliczeniowej mogą być zróżnicowane. W części zastosowań wystarczy charakterystykę wzorców zapisać w pamięci systemu, gdy nie zależą one

od żadnych czynników. Czasem konieczne może być jednak obliczenie wzorca na podstawie parametrów nowej obserwacji (np. lokalizacja determinująca poziom podświetlenia dłoni a więc odbierany profil), co znacznie podniesie całkowitą złożoność obliczeniową metody.

d) *Maszyna wektorów nośnych*

Jedną z metod z grupy uczenia maszynowego wykorzystywaną w klasyfikacji, która wyznacza hiperpłaszczyznę rozdzielającą obserwacje poszczególnych klas przechowywanych w postaci wektorów cech jest maszyna wektorów nośnych (ang. *Support Vector Machines*, SVM). Idea SVM zaproponowanej przez Vapnika [115], polega na znalezieniu optymalnej hiperpłaszczyzny, która rozdzieli obserwacje danych klas z możliwie największym i równym dla każdej grupy marginesem. W pierwotnej implementacji był to klasyfikator liniowy, lecz zastosowanie funkcji jądra (np. gaussowskiego) pozwoliło również na rozróżnianie elementów ze zbiorów nieliniowo separowalnych. Jednym z jej hiperparametrów jest wartość C , która definiuje dopuszczalną szerokość marginesów pomiędzy klasami. Im większy uzyskany margines tym większa zdolność modelu do uogólniania.

Wykorzystanie SVM do klasyfikacji póż wykorzystano m.in. w [94]. SVM wykorzystano do estymacji położenia dłoni wykonującej gest jako część dwustopniowego klasyfikatora w pracy Withana i in. [65]. Metodę zastosowano w zestawieniu porównawczym klasyfikacji póż na podstawie odbitych impulsów radiowych [100]. Rozpoznawanie grupy sześciu gestów bazujące na obrazie wideo i metodzie SVM opisali Maqueda i in. [82].

e) *Sztuczne sieci neuronowe*

Koncepcja sztucznych sieci neuronowych, (ang. *Artificial Neural Networks*, ANN) została sformułowana na podstawie analogii do działania układu nerwowego człowieka. Mimo, że ich masowe zastosowanie upowszechniło się dopiero wraz z odpowiednim rozwojem metod uczenia sieci a przede wszystkim mocy obliczeniowej komputerów po roku 2000, zostały one zaproponowane już w 1943 roku przez McCullocha i Pittsa [116]. Obecnie jest to jeden z najbardziej powszechnych modeli z grupy uczenia maszynowego. Model sztucznych sieci neuronowych jest wykorzystywany zarówno do regresji (przewidywania wartości na podstawie zmiennych) jak i klasyfikacji. W tym podpunkcie,

z racji potrzeb wykorzystania modelu rozpoznawania pól, skupiono się na drugiej z możliwości.

Wykorzystanie ANN w klasyfikacji danych jednowymiarowych polega na podaniu na wejście klasycznej sieci neuronowej wektora danych (obserwacji obiektu jakiejś klasy) x_d , którego poszczególne wartości będą wewnątrz sieci wzajemnie na siebie oddziaływać z siłą i kierunkiem zdefiniowanym przez połączenia między poszczególnymi neuronami. Oddziaływania realizowane za pomocą operacji matematycznych prowadzą do uzyskania z sieci rozpoznającej obiekty N klas odpowiedzi o prawdopodobieństwie klasyfikacji danej obserwacji do każdej z N klas.

Rozważmy sieć składającą się z jednego sztucznego neuronu. Oddziaływania sygnałów dla jednego neuronu można zapisać formułą matematyczną:

$$h(x) = \phi(w^T \cdot x_w + b), \quad (2.2)$$

gdzie w - wektor z wagami, po jednej dla każdego z wejść neuronu, x_w - wektor sygnałów wejściowych do neuronu, b - obciążenie (ang. *bias*) neuronu, ϕ - funkcja aktywacji neuronu, h - odpowiedź (wyjście) neuronu. Argument funkcji aktywacji często oznacza się symbolem z . Każdy z neuronów sieci posiada tyle wag ile wchodzi do niego wejść (z wektora danych lub z innych neuronów). W neuronach odpowiedzialnych za rozpoznawanie obserwacji najczęściej stosuje się funkcje aktywacji ϕ typu sigmoidalnego lub tangensoidalnego (płytkie sieci) lub ReLU i jego modyfikacje (ELU, SELU) (głębokie sieci). Wynikiem działania neuronu jest zatem przetworzenie zestawu wejść do jednego sygnału wyjściowego. Przykładowo, dla jednoneuronowej sieci realizującej regresję należy tak dobrać wagi i obciążenie aby odpowiedź h sieci na każdy zestaw wejściowy x_w była możliwie równa znanemu wyjściu stowarzyszonemu z x_w . Dobieranie wag i obciążeń odbywa się dzięki zastosowaniu jednej z procedur uczenia sieci. Polegają one na iteracyjnej aktualizacji wartości początkowo losowo wybranych wag i obciążeń na podstawie minimalizacji różnicy między oczekiwaną a obliczoną odpowiedzią sieci. Przykładową metodą uczenia jest algorytm wstecznej propagacji błędów (ang. *backpropagation*). Algorytmy uczenia sieci umożliwiają klasyfikację za pomocą większych struktur składających się z dużej liczby neuronów lecz proces ten wymaga wielu przykładów par: zadane wejście - spodziewane wyjście (rzędu co najmniej setek na klasę).

Topologia typowej sztucznej sieci neuronowej składa się z trzech rodzajów warstw. Warstwa wejściowa posiada liczbę neuronów odpowiadającą długości wektora danych

podawanych do sieci i służy do pobierania i przygotowania danych. Warstw ukrytych (WU) może być od jednej (płytkie sieci) do bardzo wielu (głębokie sieci). W zastosowaniach rozpoznawania obrazu buduje się modele posiadające ponad 150 różnego rodzaju warstw ukrytych [117]. Ilość warstw i liczbę neuronów w WU (jedne z hiperparametrów sieci neuronowej) należy dobrać w zależności od rodzaju danych i ewentualnych ograniczeń systemu (rozmiar pamięci, prędkość przetwarzania). Warstwa wyjściowa często składa się z ilości neuronów odpowiadającej liczbie rozpoznawanych klas. W problemach klasyfikacyjnych jako funkcję aktywacji neuronów warstwy wyjściowej stosuje się funkcję *softmax*, która bierze pod uwagę wartości wyjść wszystkich neuronów tej warstwy i oblicza prawdopodobieństwo przynależności danego sygnału wejściowego do każdej z klas, jakie przewiduje dana sieć neuronowa [118]:

$$\phi(\mathbf{z})_j = \frac{\exp(z_j)}{\sum_{k=1}^K \exp(z_k)}, \text{ dla } j=1, \dots, K, \quad (2.1.)$$

gdzie \mathbf{z} – wektor wyjść wszystkich neuronów warstwy wyjściowej przed zastosowaniem funkcji aktywacji, z_j - wyjście j -tego neuronu przed zastosowaniem funkcji aktywacji, K - ilość rozpoznawanych klas. Cechą funkcji *softmax* jest zatem, iż suma wartości wszystkich wyjść z neuronów z tej warstwy po zastosowaniu funkcji aktywacji równa się jedności.

Taki typ sieci neuronowej, w której kierunek przepływu informacji jest zawsze od warstwy wejściowej przez warstwy ukryte do warstwy wyjściowej, określa się jako *feedforward*. Klasyczną sieć neuronową, z co najmniej trzema warstwami i z nieliniowymi (nie licząc warstwy wejściowej) funkcjami aktywacji nazywa się też akronimem MLP (ang. *Multi Layer Perceptron*).

Klasyczna struktura sieci neuronowych jest często wykorzystywana w zagadnieniach klasyfikacji poszczególnych pól dłoni. Sparаметryzowany obraz z kamery podawany był na wejście MLP z jedną warstwą ukrytą i klasyfikującą 5 pól przez [90]. Sieć o podobnej topologii zastosowano do rozpoznawania 5 gestów na podstawie 16 cech z sensora EMG (ang. *Electromyography*) [119]. Futane i in. obliczali cechy otrzymanego kształtu dłoni i umieszczali je na wejściach sieci neuronowej typu General purpose Fuzzy MinMax [120]. Dla rozpoznawania gestów rąk Yang i in. wykorzystywali trajektorie ruchu i model sieci neuronowej z opóźnieniem w czasie [121].

Powstało również rozszerzenie modelu sztucznych sieci neuronowych przeznaczone do analizy obrazów. Polega ono na obliczaniu cech poszczególnych segmentów obrazu

za pomocą splotu (ang. *convolution*) z różnego rodzaju filtrami, których wynik, po kilku etapach przygotowania danych, jest analizowany przez klasyczne warstwy sieci neuronowych. Takiego typu struktury nazywane są sieciami splotowymi (ang. *Convolutional Neural Networks*, CNN). Wykorzystywane były one przy rozpoznawaniu gestów z sygnału video m.in. przez [122][123][124]. Jednowymiarowy spłot zastosowano zaś przy wykorzystaniu modelu CNN w analizie danych czasowych z czujnika gestów rejestrującego odbicie fali radiowej [100].

2.3.2. Metody klasyfikacji sekwencji

Zagadnienie rozpoznawania gestów (gdzie ważna może być zarówno sama poza jak i lokalizacja / trajektoria) polega na interpretacji całej sekwencji pakietów danych w następujących po sobie chwilach czasu (cyklach próbkowania). W tej sekcji przedstawiono wybrane metody używane do klasyfikacji sekwencji.

a) Maszyny stanów

Podstawowa wersja maszyny stanów (ang. *Finite State Machine*, FSM) jest matematycznym modelem opisu zachowania systemu poprzez tablicę dyskretnie opisanych przejść pomiędzy stanami zdefiniowanymi w jego grafie. Pierwsze sformułowanie konceptu FSM zostało zawarte w pracy McCullocha i Pittsa wprowadzającej również sztuczne sieci neuronowe [116]. System opisany przez FSM w danym momencie czasu znajduje się dokładnie w jednym spośród skończonej liczby stanów. Stan, w którym znajduje się maszyna zmienia się (tzw. przejście) pod wpływem oddziaływania informacji wejściowych do maszyny (np. wartość sygnału na fotodiodzie przekraczająca dany próg) na warunki przejścia danego stanu. Z każdym przejściem można związać także odpowiedź maszyny (wyjście). Cała maszyna zdefiniowana jest przez stan początkowy I , listę pozostałych stanów (S_i) Q , zbiór symboli/zdarzeń wejściowych (e_i) Σ_1 , zbiór symboli wyjściowych (o_i) Σ_2 oraz warunki przejścia (relacje przemiany) dla każdego z nich (λ_i) Λ . Relacja przemiany λ_i między stanami S a S' , gdzie g reprezentuje warunki pozwoleń na przejście, opisana jest następująco:

$$\lambda_i: S \xrightarrow{e[g]/\{o\}} S'. \quad (2.3)$$

Model matematyczny FSM nie przewiduje jawnie obecności liczników, toteż klasyfikacja sekwencji o N_{sek} następujących po sobie wejść o tej samej informacji (np. statyczna poza dłoni utrzymująca się przez co najmniej N_{sek} cykli próbkowania) wymaga utworzenia maszyny o co najmniej N_{sek} stanach (pomijając stan początkowy i pomocnicze). Ta niedoskonałość modelu powoduje, iż opis upływu czasu dla dłuższych sekwencji (np. dla systemów z wysoką częstotliwością próbkowania) staje się kłopotliwy. Jednak skończone maszyny stanów sprawdzają się znakomicie w zadaniach dla systemów mniej dynamicznych.

Z powodu jej formalnych ograniczeń w literaturze zaproponowano kilka modyfikacji modelu FSM. Model rozszerzonej maszyny stanów, EFSM (ang. *Extended Finite State Machine*) jest analogiczny jak dla FSM, lecz zawiera także zbiór zmiennych V , które wykorzystuje się często jako liczniki czasu lub zmienne pomocnicze [125][126]. W rozważanym przykładzie sekwencji o N_{sek} jednakowych obserwacjach, używając EFSM wystarczyłoby zdefiniować dwa stany: domyślny S_d i liczenia obserwacji S_{ob} . Ponieważ każdy ze stanów może mieć kilka warunków przejść, w stanie S_{ob} jeden z nich odpowiadałby za zwiększanie licznika aż do N_{sek} przy obserwacji wciąż tego samego wejścia do maszyny (zwinięcie graficzne i semantyczne). System EFSM składa się zasadniczo z trzech etapów cyklu zachowania: ocena wszystkich warunków przejścia, obliczenie następnego stanu i kontrola sygnałów.

Jedną z kolejnych modyfikacji FSM są periodyczne maszyny stanów PFSM (ang. *Periodic Finite State Machine*) jawnie uwzględniające upływ czasu [127]. Model ten zawiera ograniczenia czasowe oraz takie pojęcia jak czas globalny, ograniczenia zegara periodycznego i zdarzenia aktywowane czasem przez co ułatwiają one definiowanie modeli zdarzeń wolnozmiennych (modele rzadkiego czasu, ang. *sparse time model*). Sięgając do przykładu sekwencji o N_{sek} obserwacjach, również wystarczyłby do jej opisu jeden dodatkowy stan, tym razem opisany m.in. przez wymagany czas trwania cyklu.

Różne warianty maszyn stanów stosowane były zarówno w kilkudetektorowych komercyjnych czujnikach prostych gestów [35][41][69] i także opisanych w literaturze [5][44][60] jak i bardziej złożonych systemach analizy video [128].

b) *Ukryte modele Markowa*

Kolejnym przykładem modelu stosowanego do klasyfikacji sekwencji zdarzeń są Ukryte Modele Markowa (ang. *Hidden Markov Models*, HMM). Opierają się one na koncepcji łańcucha Markowa i są zdefiniowane dla czasu dyskretnego. Analizowane sekwencje muszą wykazywać się własnością Markowa czyli każda kolejna obserwacja w sekwencji nie może zależeć od poprzedniej. Nazwa metody wiąże się z faktem, iż HMM modeluje proces stochastyczny, w związku z czym jego właściwości nie są pewne, więc określa się je jako ukryte. Następujące po sobie obserwacje w analizowanej sekwencji są reprezentowane przez ukryty model Markowa a każdy z nich jest automatem skończonym (FSM). Zawiera zatem skończoną liczbę i stanów i przejść między nimi.

HMM wykorzystywano w wielu pracach nad rozpoznawaniem gestów jako metodę główną [33][71][80][129][130][131] lub referencyjną [88] na przestrzeni wielu lat.

c) *Rekurencyjne sieci neuronowe*

Popularny model do analizy sekwencji wykorzystujący sztuczne sieci neuronowe bazuje na strukturze grafu rozwijanego w czasie. Model ten, rekurencyjna sieć neuronowa (ang. *Recurrent Neural Network*, RNN) w pierwszym przybliżeniu wygląda jak klasyczna sieć *feedforward*: może posiadać warstwę wejściową, warstwy ukryte i wyjściową. Różnica polega jednak na zastosowaniu w sieci dodatkowych połączeń rekurencyjnych do analizy sekwencji o N stemplach czasowych (czyli składającej się z N obserwacji). Ich idea polega na tym, że dla stempla czasowego t danej sekwencji neuron przetwarza t -ty zestaw danych wejściowych, zaś do obliczenia swojego wyjścia uwzględnia także wyjście z $t-1$ stempla, więc i swoją wartość $\phi(z)$ przekazuje zarówno na wyjście neuronu jak i do przetwarzania $t+1$ stempla [132]. Należy zaznaczyć, że dla przetwarzania danych z każdego stempla czasowego w sekwencji używany jest ten sam neuron, czyli kolejne obserwacje (i potencjalnie różne wyjścia z poprzedzających obserwacji) oddziałują na ten sam zestaw wag i obciążeń. Wyjście pojedynczego rekurencyjnego neuronu można zapisać zatem następująco [133]:

$$y_{(t)} = \phi(x_{(t)}^T \cdot w_x + y_{(t-1)}^T \cdot w_y + b), \quad (2.4)$$

gdzie $x_{(t)}$ to obserwacja w stemplu czasowym t , w_x to zestaw wag neuronu na jego wejściu, w_y to zestaw wag neuronu na połączeniu rekurencyjnym (między wyjściem neuronu

a wejściem do niego), b to obciążenie, $y_{(t)}$ to wyjście z neuronu w stemplu czasowym t zaś ϕ to funkcja aktywacji. Wobec tego, że w każdym stemplu czasowym wyjście przekazywane jest dalej i uwzględniane aż do końca sekwencji, można uznać, iż sieć rekurencyjna posiada swego rodzaju pamięć. Neuron rekurencyjny jest zatem przykładem prostej komórki pamięci. Jednak trening sieci tego typu na sekwencjach o długościach rzędu 100 stempli czasowych jest problematyczny i bardzo powolny. W sieciach głębokich (a rozwinięcie grafu dla 100 stempli czasowych tworzy bardzo głęboką sieć) algorytmy uczenia bazujące na obliczaniu pochodnych (gradientów) często szybko przestają aktualizować wagi (ang. *vanishing gradients problem*) lub przypisują im zbyt duże wartości, podatne na błędne obliczanie wyjścia lub przestające mieścić się w zakresie zmiennej (ang. *exploding gradients problem*) [134]. Obok metod wsparcia uczenia zapobiegających problemom z gradientami (np. odpowiednie funkcje aktywacji, obcinanie gradientu) zaproponowane zostały przede wszystkim bardziej zaawansowane struktury komórki pamięci.

Komórka LSTM (ang. *Long Short-Term Memory*), zaproponowana przez Hochreitera i Schmidhubera [135], zawiera specjalne bramki (zapominania, wejściowa i wyjściowa), odpowiedzialne za interpretowanie i decyzyjowanie przepuszczania tylko ważnej informacji do uwzględnienia w następnych stemplach czasowych. Przekazuje ona tę informację w postaci nie jednej ale dwóch wartości. Są to stan krótkotrwały $h_{(t)}$ oraz stan długotrwały $c_{(t)}$.

Komórką pamięci o prostszej budowie niż LSTM jest GRU (ang. *Gated Recurrent Unit*) opisana przez Chunga i in. [136]. Tak jak neuron rekurencyjny, przekazuje ona dalej tylko jeden wektor stanu. Posiada jedną bramkę mniej (brak bramki wyjściowej) lecz zawiera dodatkowy kontroler bramki. W wynikach różnych badań zaprezentowano, iż komórki GRU wykazują w co najmniej porównywalną użyteczność w uczeniu sekwencji jak LSTM [137][138] oraz, iż obydwie dla długich sekwencji dają znacznie lepsze rezultaty niż zwykła komórka pamięci [136]. Ze względu na swoją prostszą strukturę czas uczenia GRU (lub liczba epok) jest krótszy niż dla LSTM i również dlatego stają się one popularne w różnych zastosowaniach.

Celem nauczania sieci rekurencyjnej dużej liczby sekwencji możliwe jest umieszczanie wielu komórek pamięci w jednej warstwie jak i tworzenie struktur rekurencyjnych wielowarstwowych. W przypadku zagadnienia klasyfikacji, tak jak w klasycznych sieciach neuronowych, należy w warstwie wyjściowej sieci zastosować

funkcję aktywacji *softmax*. Jednak jako wynik klasyfikacji całej sekwencji interpretuje się tylko odpowiedź sieci z ostatniego stempla czasowego (tryb sekwencji do wektora).

Dzięki zastosowaniu złożonych komórek pamięci i metod uczenia sieci rekurencyjnych znalazły one zastosowanie w tłumaczeniu języków, przetwarzaniu mowy w tekst i odwrotnie czy w przewidywaniu przebiegów [139][140]. Często w połączeniu z sieciami do analizy obrazów (CNN) [122][124][141] lub w połączeniu z innymi metodami [88][130][137][142] sieci rekurencyjne wykorzystuje się do rozpoznawania gestów jako model klasyfikacji sekwencji.

2.4. Miary efektywności metod klasyfikacji

Sekcja ta przedstawia opis podstawowych miar wykorzystywanych do opisu i porównywania jakości klasyfikatorów projektowanych w dalszej części rozprawy.

- Dokładność klasyfikacji

Miarę dokładności (ang. *accuracy*) klasyfikacji przypadków należących do klasy j określa się jako stosunek liczby poprawnie zaklasyfikowanych przypadków należących do tej klasy, n_{j_pop} , do całkowitej liczby przypadków tej klasy j , n_{j_sum} :

$$A_j = \frac{n_{j_pop}}{n_{j_sum}} \times 100\%. \quad (2.5)$$

Można również wyrazić średnią dokładność klasyfikatora, w odniesieniu do rozpoznawania przypadków wszystkich K klas, których został on nauczony:

$$A = \frac{\sum_{j=1}^K A_j}{K} \times 100\%, \quad (2.6)$$

gdzie: A_j to dokładność klasyfikacji przypadków należących do klasy j .

- Czulość i swoistość

Czulość testu diagnostycznego opisuje jego zdolność do poprawnego wykrycia pewnej badanej cechy. Jest to stosunek wyników prawdziwie dodatnich (ang. *True Positive*, TP) do sumy prawdziwie dodatnich i fałszywie ujemnych (ang. *False Negative*, FN). Swoistość testu diagnostycznego opisuje zdolność testu do wykrywania braku danej cechy i definiuje się ją przez stosunek wyników prawdziwie ujemnych (ang. *True Negative*, TN), do sumy prawdziwie ujemnych i fałszywie dodatnich (ang. *False Positive*, FP).

- Krzywa ROC

Przebieg krzywej ROC (ang. *Receiver Operating Characteristic*) otrzymuje się przez wykreślenie *True Positive Rate* (czułości testu) w funkcji *False Positive Rate* (1-swoistości) dla różnych wartości progowych. Wartości progowe dotyczą wielkości prawdopodobieństwa, powyżej którego sygnał wejściowy zaklasyfikowany jest do kategorii pozytywnie rozpoznanych (a poniżej którego, do negatywnie rozpoznanych). Im bardziej wypukła krzywa, tym klasyfikator jest lepszy. Krzywa ROC idealnego klasyfikatora jest prostokątna i przechodzi przez punkt (0,1) i oznacza, że właściwie zaklasyfikował on wszystkie sygnały wejściowe. Jednym z parametrów krzywej ROC jest pole pod wykresem (ang. *Area Under the ROC, AUROC*). Określa ono prawdopodobieństwo, iż rozważany model będzie w stanie wyżej ocenić losowy przypadek klasy pozytywnej niż losowy przypadek klasy negatywnej.

- Tablica pomyłek

Jest to tabela pokazująca dokładność klasyfikatora w rozpoznawaniu przypadków należących do poszczególnych klas. Jej liczba rzędów i kolumn odpowiada liczbie rozważanych klas. Suma wartości w rzędach odpowiada ilości przypadków prawdziwie przynależących do danej klasy. Jeżeli są wyrażone w procentach, suma wartości każdego rzędu równa się 100%. W kolumnach zaś znajdują się licznosci przypadków przewidzianych przez klasyfikator do danej klasy (opisany przez numer rzędu tablicy). Wartości na przekątnej tabeli to czułość klasyfikatora w rozpoznawaniu przypadków danej klasy. Wartość komórki w rzędzie r i kolumnie k mówi o liczbie przypadków klasy opisanej przez rząd r lecz zaklasyfikowanych do klasy opisanej przez kolumnę k .

2.5. Podstawowy zbiór póz dłoni

Jak zdefiniowano w celach niniejszej rozprawy, w jej ramach analizowane są możliwości rozpoznawania gestów za pomocą zaproponowanego kilkuelementowego czujnika nie bazującego na przetworniku matrycowym (np. kamera RGB). Z tego względu, w zależności od jego budowy i zasady działania, jego zdolności rozpoznawania gestów będą, w stosunku do możliwości dawanych przez np. analizę obrazu, ograniczone. Celem zdefiniowania części wymagań technicznych i funkcjonalnych projektowanego czujnika gestów ustalić należy jakie pozy dłoni powinien on rozróżniać. W początkowych badaniach dokonano rozróżnienia na pozy dłoni o palcach zwartych

FJ (ang. *fingers joined*) i rozłożonych FS (ang. *fingers sparsed* lub *spread*) [47]. Ze względu na fizjologiczną niewygodę wykonywania póz o trzech i czterech palcach rozpostartych w porównaniu z pozostałymi pozami, nie rozważano ich rozpoznawania w dalszych badaniach. Zaproponowano zatem książkę kodową określającą pozy dłoni będące przedmiotem dalszych analiz (Tabela 2.1). Ułożenia palców dłoni w rozważane pozy zostały przedstawione na Rys. 2.6.

Tabela 2.1 Pozy dłoni rozważane do wykrywania przez liniowy sensor gestów.

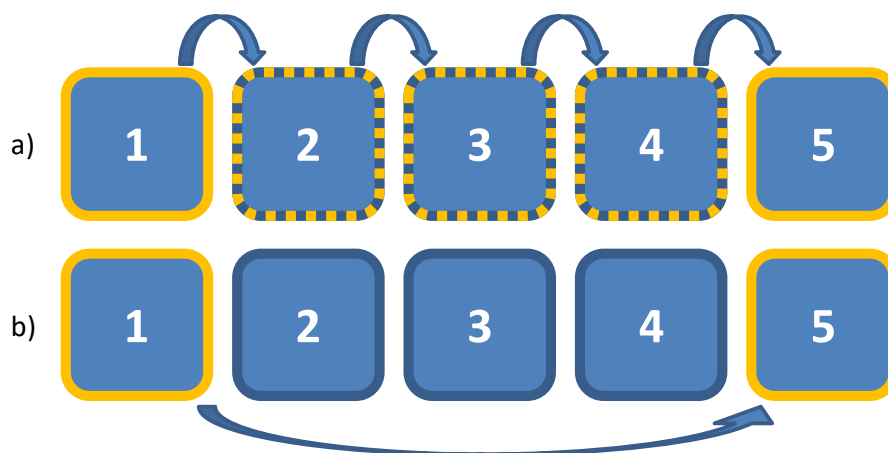
Gest	Kod
1 palec	1FS
2 palce rozłożone	2FS
Wąska powierzchnia (2 palce złożone)	2FJ
Szeroka powierzchnia (4 palce złożone)	4FJ



Rys. 2.6 Pozy dłoni rozważane do wykrycia przez liniowy optyczny sensor gestów.

2.6. Projekt zbioru akcji do obsługi urządzeń mobilnych

W tej sekcji podjęto problem oszacowania rodzaju i liczby akcji, za pomocą których bezdotykowa nawigacja użytkownika po mobilnych systemach operacyjnych ma szansę być szybka i precyzyjna. Duża ilość akcji wywoływanych na inne sposoby pozwala bowiem na wyzwalanie różnych funkcji aplikacji mobilnej. Może to zostać wykorzystane jako odciążenie użytkownika np. od częstych ruchów prowadzących jedynie do zmiany podświetlenia kontrolki graficznego interfejsu użytkownika, którą ma on zamiar aktywować co jest realnym zagrożeniem dla tego typu interfejsów (Rys. 2.7). Im więcej różnych gestów będzie w stanie rozpoznawać czujnik gestów tym więcej akcji będzie mogło być wywoływanych w osobny sposób, czyniąc interakcję użytkownika z urządzeniem bardziej efektywną.



Rys. 2.7 Przykład zalet interfejsów obsługujących wiele gestów. a) Zmiana podświetlenia aktywnej kontrolki GUI między guzikiem 1 a 5 dla interfejsu obsługującego jedynie akcję „następna kontrolka” wymaga wykonania akcji pośrednich (w sumie 4 akcje). b) Analogiczna zmiana z wykorzystaniem interfejsu obsługującego dodatkową, wyzwalaną w inny sposób, akcję „4 kontrolki do przodu” pozwala na redukcję wykonywanych przez użytkownika akcji (w sumie 1 akcja) a więc szybszy dostęp do funkcji aplikacji.

Interfejsy graficzne użytkownika dla komputerów stacjonarnych czy laptopów (np. Windows XP, Linux Ubuntu) o dużych (w stosunku do urządzeń mobilnych) wyświetlaczach, posiadających klawiaturę oraz mysz projektowane są w konwencji WIMP (ang. *windows-icons-menus-pointers*) [143]. Oznacza to, iż są one przystosowane do wyświetlania dużej ilości treści, precyzyjnej manipulacji (mysz) i bardzo rozwiniętego interfejsu edycyjnego (klawiatura). Interfejsy dla urządzeń mobilnych takich jak smartfony czy okulary multimedialne (bazujące np. na systemach Android czy iOS) charakteryzują się konwencją interakcji post-WIMP [144]. Wyróżnia się ona przystosowaniem do innych niż dla komputera metod wprowadzania danych (np. dotykowy ekran, ograniczony zestaw komend wprowadzanych za pomocą guzików, poleceń głosowych, gestów) poprzez interfejs o dużych graficznie elementach, nie wymagających tak precyzyjnej nawigacji jak przy użyciu myszy komputerowej. Z tego powodu urządzenia post-WIMP nie są jednak wygodne do edycji i wprowadzania dużej ilości danych tekstowych. Interakcja bezdotykowa, na przykład za pomocą gestów ma tą zaletę, iż nie potrzebuje aby na ekranie urządzenia wyświetlane były choćby w postaci guzików wszystkie obsługiwane w danym momencie przez aplikację funkcje. Ich wyzwalanie odbywać się może bowiem przez wykonanie gestu a nie naciśnięcie guzika zajmującego miejsce na wyświetlaczu. W kontekście typów kontrolki GUI używanych w aplikacjach dedykowanych dla urządzeń mobilnych w niniejszej analizie rozważono w szczególności akcje związane z obsługą: potwierdzania / wybierania danej kontrolki, ukrytych menu (ang. *pop-up menu*), zmianę kart (ang. *tab*) oraz zmianę

podświetlenia aktywnej kontrolki i akcje z nimi powiązane. Rozważone zostały także akcje możliwe do wykorzystania w obsłudze różnego rodzaju multimedialnych.

Projektowanie zbioru akcji dla obsługi urządzeń mobilnych wykonano na podstawie przeprowadzonego przeglądu taksonomii gestów oraz uwzględniając charakter aplikacji dla systemów mobilnych. W tym celu wykorzystano zbiory zadań podzielone na sześć kategorii: *transformację*, *symulację*, *wybieranie*, *zaznaczanie*, *edycję* i *menu* opracowane przez Piumsomboona i in. [145] oraz Chana i in. [146]. W pierwszej z tych prac, w ramach wymienionych kategorii, autorzy zaproponowali w sumie 40 różnych zadań wykonywanych w kontekście interakcji z rozszerzoną rzeczywistością (ang. *Augmented Reality*, AR). W drugiej pracy wyróżniono 34 zadania realizowane przez mikrogesty wykonywane jedną ręką. Wynikiem adaptacji przytoczonych zbiorów zadań jest zbiór 26 akcji pogrupowanych według czterech z wymienionych kategorii. Z przyczyn opisanych w charakterystyce post-WIMP odrzucono bowiem kategorie *edycji* i *zaznaczania*. Na 26 akcji składa się 18 wybranych spośród zdefiniowanych przez Piumsomboona lub Chana oraz 8 zaproponowanych przez autora niniejszej rozprawy. Kluczem wyboru i motywacją dla powstania nowych propozycji była wygoda obsługi aplikacji dla urządzeń mobilnych. Zaproponowany zbiór akcji prezentuje Rys. 2.8.



Rys. 2.8 Zaproponowany zbiór akcji podzielonych na 4 kategorie. Akcje bez podkreślenia zaczerpnięto z opracowań literaturowych zaś akcje podkreślone zostały zaproponowane przez autora rozprawy doktorskiej.

Wybrane akcje kategorii *transformacja* mogą dotyczyć operacji wykonywanych na obrazach. Kategoria *symulacja* zaś odnosić się może do odtwarzania zarówno plików wideo jak i audio. Akcje kategorii *wybieranie* zostały rozszerzone o krokowe substytuty operacji przewijania (ang. *scroll*) w postaci przełączania kontrolerek (lub list) o kilka zamiast tylko co jedną oraz przełączanie kart aplikacji. Do wybranych akcji kategorii *menu* dodano akcję odświeżania i opuszczenia aplikacji.

Z opisu opracowania Piumsomboona wynika, iż każde okno aplikacji obsługiwanej za pomocą zdefiniowanego zbioru akcji powinno być jednoznacznie przypisane do jednej z kategorii. Oznacza to, że gesty stowarzyszone z akcjami mogą się powtarzać, ale tylko jeśli dotyczą akcji z różnych kategorii. W niniejszej rozprawie założono, iż każda akcja w zbiorze powinna być wyzwalana innym gestem.

Ze względu na potencjalne trudności w opracowaniu interfejsu bezdotykowego zdolnego do obsługi ponad 30 lub 40 gestów (jak w przytoczonych pracach) oraz ze względu na możliwe trudności z zapamiętaniem takiej liczby akcji przez użytkownika zdecydowano o propozycji ograniczenia wielkości powstałego zbioru. Nie rozważając zatem obsługi odtwarzania głosu (akcje 10, 11, 12), wykluczając jedną z dodatkowych opcji szybszego przełączania kontroltek (np. akcje 18, 19) oraz wyjście z programu (akcja 26) można oszacować liczbę akcji potrzebną do wygodnej i precyzyjnej obsługi aplikacji mobilnej na 20. Zadaniem badawczym jest zatem opracowanie czujnika gestów zdolnego do rozróżniania co najmniej 20 gestów dłoni.

Zaznaczyć należy, iż z każdą ze zdefiniowanych akcji związany będzie gest typu dyskretnego, czyli funkcja aplikacji zostanie wyzwolona po zakończeniu gestu. Istnieją jednak również kontrolki (np. pasek przeszukiwania, ang. *seek bar*), których obsługa jest bardziej wygodna za pomocą gestu ciągłego (reakcja interfejsu w trakcie wykonywania gestu) lecz możliwości ich obsługi zostaną omówione w dalszej części rozprawy.

2.7. Podsumowanie rozdziału

W niniejszym rozdziale przedstawione zostały obecne w literaturze rozwiązania optycznych czujników gestów z podziałem na grupy różniące się ilością elementów rejestrujących światło. Dostrzeżono, iż wśród metod bazujących na czujnikach maksymalnie kilkudetektorowych (gdzie maksymalna złożoność czujnika to maczyca 4x4 detektory), pod względem największej ilości rozpoznawanych gestów dyskretnych wyróżnić można pasywny czujnik Kaholokuli (10 gestów) [73], sensor złożony z czujników zbliżeniowych Kratza (9 gestów) [72] oraz aktywny czujnik Withany (do 8 gestów) [65]. Zatem na podstawie niedużej liczby gestów rozpoznawanych przez najlepsze pod tym względem czujniki optyczne (do 10) oraz dzięki propozycji minimalnej wymaganej liczby gestów do obsługi przez interfejs dla urządzeń mobilnych (20) zdecydowano o projekcie typu czujnika i dedykowanych mu metod klasyfikacji gestów spełniających te wymagania. Zaobserwowano także, że część czujników jedno- lub kilkudetektorowych,

które potrafią rozróżnić pozy dłoni bazuje na skanowaniu dłoni przemieszczającej się przed czujnikiem (pozy dynamiczne) [44], lecz rozpoznawano także kilka (do trzech) statycznych póz dłoni [65]. Rozróżnianie przez czujnik póz zarówno statycznych jak i dynamicznych jest istotne, gdyż każda z nich dodaje stopień swobody do typu rozpoznawanych gestów, co szybko może zwiększyć ogólną liczbę gestów dyskretnych rozpoznawanych przez czujnik. Przykładowo rozpoznawanie zarówno kierunku przesunięcia dłonią i jej pozy w trakcie wykonywania ruchu oraz wykrywanie tej samej pozy lecz statycznej (brak ruchu dłoni) pozwala na zdefiniowanie już 3 gestów dla jednej pozy, co dodatkowo można jeszcze zróżnicować (dla przesunięć) choćby tempem wykonywania ruchu. Z tego względu zdecydowano o projekcie czujnika zdolnego do detekcji co najmniej 4 statycznych póz dłoni, na bazie kombinacji których zdefiniowana zostanie baza gestów. Ponadto, tylko w niewielu spośród zaprezentowanych pozycji literaturowych autorzy rozważali możliwość rozpoznawania gestów zarówno dyskretnych jak i ciągłych za pomocą tego samego urządzenia [59][60]. W badaniach zamieszczonych w niniejszej rozprawie rozważone zostaną zatem w szerszym zakresie możliwości projektowanego czujnika zarówno w detekcji gestów dyskretnych jak i jego zdolnościach do obsługi gestów ciągłych. Dla kilku spośród czujników opisanych w prezentacji stanu wiedzy przeprowadzono badania związane z ich dokładnością detekcji gestów w różnych warunkach oświetlenia [71][73]. Autorzy proponowanych rozwiązań nie rozważali jednak osobnych trybów pracy urządzenia (np. poprzez załączanie diod LED), które mogłyby zależeć od odmiennych warunków pracy czujnika (dla czujników optycznych: poziomu natężenia światła otoczenia). Z tego względu w zakresie niniejszej rozprawy rozważone zostaną możliwości działania optycznego czujnika gestów w różnych warunkach oświetlenia, dostosowując tryb pracy do poziomu światła otoczenia i potencjalnie pozwalających na oszczędność energii związanej ze świeceniem diod LED.

Następnie omówiono schemat systemu rozpoznawania gestów, GRS, możliwy do zaadaptowania w różnych rozwiązaniach, gdyż niezależnie od rodzaju czujnika i stopnia jego złożoności zdefiniowane w jego ramach fazy będą realizowały tę samą funkcjonalność. Adaptacja przez autora rozprawy schematu GRS zaproponowanego przez Berman i Sterna [34] polegała głównie na wyraźnym rozdzieleniu procesu rozpoznawania gestu na fazę analizy pozy i klasyfikacji całego gestu. Z tego powodu, w pierwszej kolejności w niniejszej sekcji wymienione zostały wybrane metody klasyfikacji

danych stacjonarnych, często stosowane w literaturze także przy zagadnieniu rozpoznawania póż dłoni. Ze względu na przeprowadzone w fazie eksperymentów i prototypowania obserwacje kształtu sygnału z czujnika gestów, zbudowanego i wykorzystywanego na potrzeby tej rozprawy, jako jedną z metod wybrano dopasowywanie wzorców. Wybrano także w celach porównawczych drugą metodę, sztuczne sieci neuronowe (ANN) jako reprezentanta metod z grupy uczenia maszynowego. W następnej kolejności zaprezentowano wybrane metody klasyfikacji całych sekwencji obserwacji, z powodzeniem wykorzystywane w literaturze w zagadnieniach rozpoznawania gestów. Jako klasyfikator na potrzeby niniejszej rozprawy wybrano model rekurencyjnych sieci neuronowych (RNN).

Rozdział zawiera także opis metod wykorzystywanych w dalszej części rozprawy do opisu jakości zaprojektowanych klasyfikatorów i zestawiania ich ze sobą. Ponadto wymieniono także cztery rozważane w rozprawie pozy dłoni, na które składają się pozy zwarte: 1 palec, 2 palce złożone i 4 palce złożone (o kodach kolejno 1FS, 2FJ, 4FJ) oraz poza rozpostarta: 2 palce rozłożone (2FS). W dalszych pracach skupiono się zatem na dwóch aspektach. Po pierwsze, aby zaproponować konstrukcję geometryczną sensora (np. ilość, rozłożenie i pole widzenia poszczególnych elementów światłoczułych) pozwalającą na rozróżnianie póż o rozpostartych i zwartych palcach dla dłoni znajdującej się w zdefiniowanym obszarze operacyjnym czujnika. Po drugie, w kontekście danej konstrukcji czujnika, należy zweryfikować czy któraś z wybranych metod klasyfikacji zwartych póż dłoni będzie charakteryzowała się porównywalną dokładnością klasyfikacji w stosunku do metod wymienionych w literaturze oraz czy będzie konkurencyjna pod względem wymagań obliczeniowych. Niewysoka złożoność obliczeniowa metody rozpoznawania póż pozwoli bowiem przykładowo na rozważenie dwustopniowego klasyfikatora gestów mogącego, dzięki wysokiej częstotliwości próbkowania, analizować sekwencje póż i dokładną trajektorię przemieszczającej się dłoni.

Rozdział kończy analiza wymagań bezdotykowej obsługi systemu mobilnego pod kątem maksymalizacji wygody użytkownika i precyzji działań. Oszacowana została minimalna liczba akcji (20), która, zdaniem autora niniejszej rozprawy, pozwala na wygodną obsługę aplikacji i kontrolek graficznego interfejsu użytkownika systemu mobilnego. Przedstawiona analiza oraz wymaganie aby każda akcja wywoływana była w osobny sposób wskazuje na uzasadnione zapotrzebowanie na bezdotykowe interfejsy rozróżniające co najmniej 20 gestów.

3. OPTYCZNY CZUJNIK GESTÓW O LINIOWEJ BUDOWIE

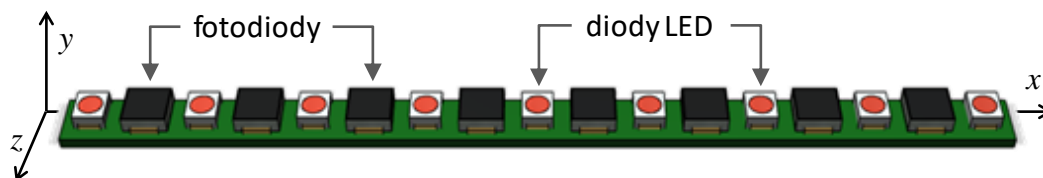
Niniejszy rozdział prezentuje ewaluację koncepcji optycznego czujnika gestów o budowie liniowej wykorzystywanego w badaniach nad metodami interakcji z urządzeniami mobilnymi. W pierwszej części rozdziału opisano zasadę działania czujnika optycznego uwzględniając zróżnicowane warunki oświetlenia panujące w otoczeniu poprzez propozycję dwóch trybów pracy: aktywnego i pasywnego. Następnie zaprezentowano wykonane na potrzeby ewaluacji koncepcji i zasady działania dwie realizacje prototypów czujnika optycznego. W dalszej kolejności, czujnik o liniowej budowie został opisany za pomocą szeregu parametrów, zarówno zdefiniowanych w literaturze przez innych badaczy jak i zaproponowanych przez autora niniejszej rozprawy. Przedstawiono metody ewaluacji właściwości sensora: stanowisko pomiarowe wraz z fizyczną realizacją czujnika oraz utworzone środowisko badań symulacyjnych oparte na metodzie Monte Carlo. W następnej sekcji umieszczono zestawienie wyników badań eksperymentalnych i symulacyjnych określających właściwości różnych konfiguracji sensora o liniowej budowie różniących się rozmieszczeniem i ilością elementów optoelektronicznych. Rozdział kończą rozważania dotyczące możliwości zastosowania opisywanego czujnika optycznego jako interfejsu gestów ciągłych.

3.1. Koncepcja i zasada działania czujnika gestów o liniowej budowie

Rozważenie koncepcji efektywnego sposobu komunikacji człowieka z urządzeniami mobilnymi za pomocą sensora gestów o niskiej złożoności konstrukcyjnej doprowadziło do sformułowania następujących wymagań branych pod uwagę przy projektowaniu sensora:

- kształt i rozmiar sensora pozwalający na umieszczenie w urządzeniach mobilnych,
- projekt sensora pozbawionego matrycy o wielu elementach rejestrujących gest celem umożliwienia szybszego działania dzięki przetwarzaniu mniejszej ilości danych,
- przeznaczenie do interakcji za pomocą jednej dłoni w polu bliskim (do 5 cm) celem użycia czujnika jako urządzenia do osobistej, nie rzucającej się w oczy, nawigacji,
- minimalizacja poziomu zużycia energii i zasobów obliczeniowych celem przystosowania sensora do pracy w urządzeniach mobilnych,
- maksymalizacja ilości i rodzajów obsługiwanych gestów w stosunku do złożoności technicznej sensora i złożoności modelu rozpoznawania gestów.

W odpowiedzi na powyższe czynniki zaproponowano konstrukcję czujnika złożonego z elementów optoelektronicznych: fotodiod (rejestrujących światło) oraz diod LED (będących źródłami światła), umieszczonych w linii, na wspólnej płycie PCB (ang. *Printed Circuit Board*) (Rys. 3.1).



Rys. 3.1 Schemat koncepcji liniowego optycznego sensora gestów.

Liniowe rozmieszczenie elementów optoelektronicznych umożliwia umieszczenie sensora wzdłuż obudowy smartfona, tabletu lub wzdłuż oprawki bocznej okularów multimedialnych. Wybór elementów optycznych, jako typu przetwornika, był podyktowany głównie hipotetyczną możliwością wykorzystania obecności światła otoczenia o wysokim natężeniu do rozważenia dodatkowego trybu pracy. Kolejnym czynnikiem była potencjalna możliwość dostrojenia części właściwości sensora w sposób mechaniczny stosując przesłony optyczne modyfikujące efektywną funkcję czułości kątowej elementów optoelektronicznych. Zdecydowano o zastosowaniu podczerwieni (ang. *infrared*, IR), jako zakresu operacyjnego nadajnika i odbiornika, która jest standardowo używana przez konstruktorów podobnego rodzaju czujników optycznych [53][60][65] (zakres długości fali ok. 900 - 1000 nm).

Jedną z informacji na temat stanu środowiska zewnętrznego, którą może przechwycić czujnik optyczny o liniowej budowie jest *profil natężenia światła* rejestrowany przez fotodiody rozłożone wzdłuż osi x . Zasada działania liniowego sensora gestów może zmieniać się w zależności od warunków otoczenia sensora. W pierwszej kolejności, im większe natężenie światła padającego na powierzchnię aktywną elementu światłoczułego, E , tym większy prąd fotodiody lub wyższe napięcie wyjściowe na wzmacniaczu transimpedancyjnym (układ zamieniający prąd na napięcie). Ponieważ sensor złożony jest z elementów optycznych, jego sygnał wyjściowy będzie podatny na poziom natężenia światła o dużej zawartości podczerwieni w paśmie emisyjnym źródła światła, w praktyce głównie światła słonecznego lub z diod LED czujnika. Z tego powodu zdefiniowano dwa podstawowe tryby pracy sensora:

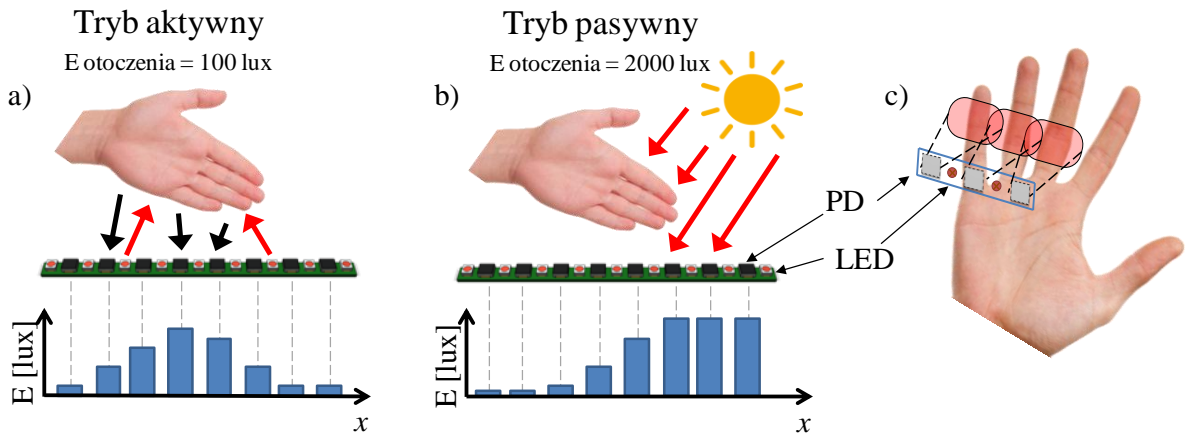
- *pasyny* - kiedy poziom natężenia światła otoczenia o dużej zawartości IR jest wysoki (więc wysoki jest poziom sygnału na fotodiodach), diody LED powinny być wyłączone, a dłoń która może znaleźć się w pobliżu sensora zablokuje światło zewnętrzne przed padaniem na część fotodiod (poziom ich sygnału wyjściowego zmniejszy się),
- *aktywny* - kiedy poziom natężenia światła otoczenia jest niski (więc niski jest poziom sygnału na fotodiodach), więc diody LED powinny być włączone, a dłoń która może znaleźć się w pobliżu sensora odbije światło emitowane przez diody LED, które padnie na część fotodiod (poziom ich sygnału wyjściowego wzrośnie).

Zatem kształt *profilu natężenia światła* rejestrowany przez sensor o budowie liniowej hipotetycznie będzie różny w zależności od trybu pracy i warunków otoczenia dla tej samej lokalizacji obiektu (np. dłoni) znajdującego się w polu widzenia sensora (Rys. 3.2a, b).

Tryb pasyny jest więc wykorzystaniem zewnętrznych źródeł światła i potencjalnym trybem energooszczędnym w stosunku do trybu aktywnego, gdzie ze względu na niski poziom natężenia światła otoczenia sensor potrzebuje uruchomić kosztowne energetycznie diody LED i doświetlić dłoń wykonującą gest. W trybie aktywnym, celem zmniejszenia zużycia energii przez układ, diody LED powinny być zapalane okresowo (z wyznaczoną częstotliwością próbkowania) oraz jedynie na czas potrzebny na przeprowadzenie próbkowania przez układ analogowo cyfrowy.

Czujnik z racji swojej liniowej konstrukcji i przy roboczym zmarginalizowaniu wpływu pola widzenia w płaszczyźnie xz będzie obserwował swego rodzaju przekrój

z przestrzeni przed nim obecnej w osi x . Ponadto, jeżeli każda fotodioda będzie posiadała ograniczone pole widzenia również w płaszczyźnie yz , w pewnym zakresie odległości od sensora będą one selektywne, rejestrując fragment przestrzeni niedostępny dla pozostałych fotodiod. Obydwie właściwości zilustrowano poglądowo na Rys. 3.2c.



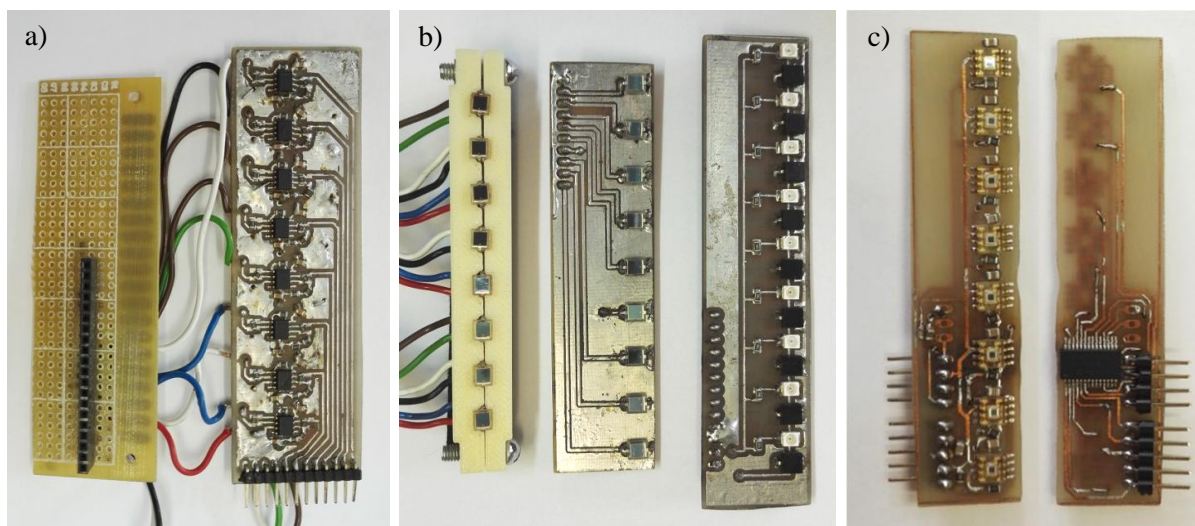
Rys. 3.2 Profile natężenia światła rejestrowane przez czujnik o budowie liniowej dla dwóch przykładowych poziomów natężenia światła otoczenia i dedykowanych im trybów pracy: a) aktywnego i b) pasywnego. c) Dualny charakter sensora: obserwacja profilu natężenia światła (przekroju przestrzeni) oraz selektywne widzenie pojedynczych fotodiod.

Kolejnym istotnym czynnikiem konstrukcyjnym sensora jest jego szerokość. Zakładając częściową selektywność fotodiod sensora i jego pożądaną zdolność do rozróżniania pól dłoni, powinien on być w stanie postrzegać obiekty o rozmiarach rzędu standardowej szerokości ludzkiej dłoni.

3.2. Realizacje liniowego sensora gestów

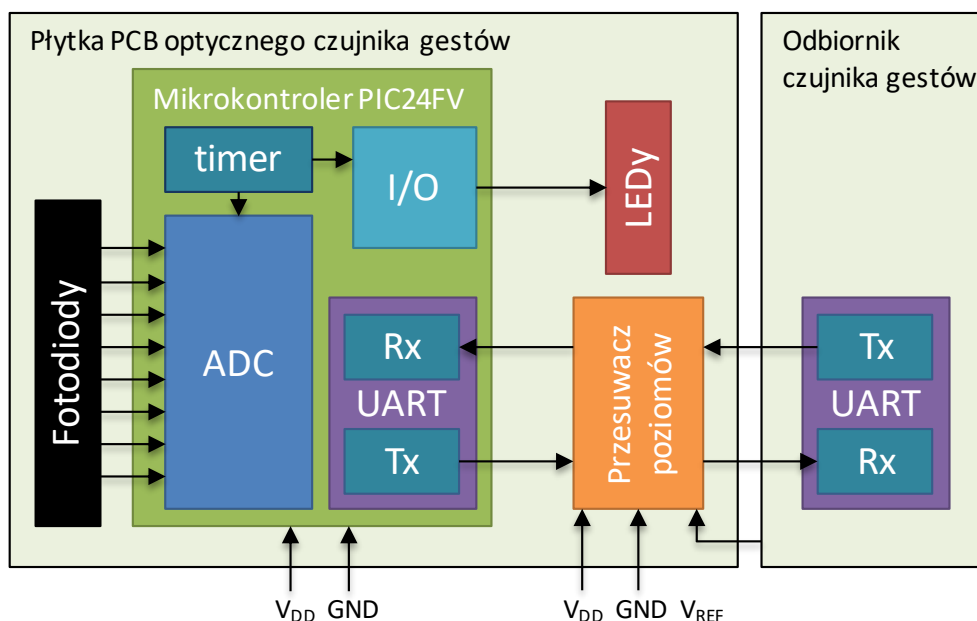
W trakcie postępów przeprowadzanych badań oraz weryfikacji zasady działania i możliwości liniowego sensora optycznego powstał szereg prototypów opracowanych dla badań eksperymentalnych, które można pogrupować w dwie realizacje:

- pierwsza - z fotodiodami wymagającymi osobnego układu wzmacniającego i transimpedancyjnego (TI) (Rys. 3.3a) oraz wymiennymi modułami z linijką fotodiod lub z linijką fotodiod oraz diod LED (Rys. 3.3b),
- druga - z fotodiodami innego typu (z wbudowanym wzmacniaczem i układem TI w obrębie jednego elementu), diodami LED, również ułożonymi w linii, oraz z mikrokontrolerem (Rys. 3.3c).



Rys. 3.3 Prototypy pierwszej (a, b) i drugiej (c) realizacji liniowego optycznego sensora gestów.
 a) Płytkę ze wzmacniaczami operacyjnymi do fotodiod (prawa część) oraz gniazdo do podłączenia różnych wersji czujnika optycznego (lewa część). b) Wymienne końcówki różnych wersji czujnika optycznego o liniowej budowie: fotodiody zakresu widzialnego (układy z lewej i środkowy), fotodiody podczerwone i diody LED (układ z prawej).
 c) Płytkę z fotodiodami zakresu widzialnego z wbudowanymi wzmacniaczami operacyjnymi (lewa część – widok górnej strony) oraz mikrokontrolerem (prawa część – widok dolnej strony).

Przy realizacji prototypów drugiej realizacji liniowego sensora gestów zdecydowano o integracji elementów optycznych oraz mikroprocesora w obrębie jednej płytki drukowanej (PCB). Takie rozwiązanie umożliwia sterowania pracą układu (zapalenie diod LED), rejestracji sygnałów (próbkiwanie) i ich przetwarzania lub przesyłania w postaci cyfrowej na inne urządzenie. Programowalny mikroprocesor czyni urządzenie samowystarczalnym, przynajmniej do pewnego stopnia złożoności implementowanego modelu rozpoznawania gestów. Dlatego niezależnie od geometrycznej konfiguracji elementów sensora i jego właściwości mógł zostać zdefiniowany schemat blokowy układu czujnika wraz z wyszczególnionymi funkcjami mikrokontrolera (Rys. 3.4).



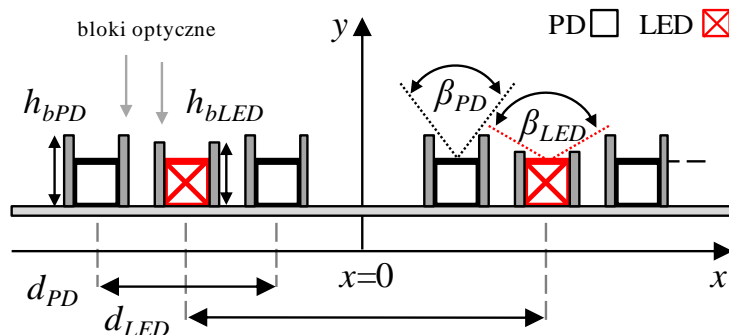
Rys. 3.4 Schemat blokowy płytki PCB i wykorzystanych bloków funkcjonalnych mikrokontrolera.

Zdecydowano o wyborze modelu mikrokontrolera PIC24FV16KA302 produkcji firmy Microchip, zasilanego napięciem o poziomie do 5 V i wyposażonego w przydatne z punktu widzenia działania sensora komponenty. W zależności od ustawień mikrokontroler może posiadać między innymi 13 wejść analogowych z 12 bitowym przetwornikiem ADC (ang. *Analog/Digital Converter*), 24 porty cyfrowe (porty wejścia/wyjścia, ang. *input/output, I/O*) czy, do komunikacji z układem, interfejs szeregowy UART (ang. *Universal Asynchronous Receiver-Transmitter*). Włączanie diod LED wykonywane było za pomocą wyjścia cyfrowego sterującego kluczem tranzystorowym doprowadzającym zasilanie do diod. Ponadto, na płytce PCB zintegrowano również układ dopasowania poziomów zasilania (ang. *level shifter*) aby można było podłączyć sensor do urządzeń sterowanych innym poziomem napięcia zasilania.

Posiadając wyjściową konstrukcję sensora optycznego o liniowej budowie zadaniem badawczym było zdefiniowanie i określenie użytecznego zakresu wartości parametrów technicznych sensora, które zapewnią konfigurację czujnika zbliżoną do optymalnej pod kątem jego efektywności energetycznej, funkcjonalnej, zdefiniowanych wytycznych i pozwalających na realizację postawionych założeń.

3.3. Parametryzacja i wymagania dla liniowego sensora gestów

Parametryzację sensora należy rozpocząć od definicji podstawowych wielkości opisujących jego geometryczną charakterystykę. Niech rozkład elementów optoelektronicznych na płycie PCB będzie opisywany za pomocą parametrów d_{PD} oraz d_{LED} określających odstęp pomiędzy sąsiadującymi elementami optoelektronicznymi tego samego typu. W rozważanych konfiguracjach diody LED odsunięte są od najbliższej fotodiody na odległość $d_{PD}/2$. Elementy optoelektroniczne umieszczone są na płycie symetrycznie względem punktu $x=0$ (Rys. 3.5). W ramach niniejszej rozprawy doktorskiej, za sensor o elementach rozłożonych uważa się taki, który spełnia nierówność: $d_{PD} \geq 5a$, gdzie a określa długość boku powierzchni aktywnej fotodiody wzdłuż osi x . Pole widzenia (ang. *Field Of View*, FOV) elementów optoelektronicznych można opisać przez kąty β_{PD} oraz β_{LED} regulowane za pomocą bloków optycznych o odpowiedniej wysokości (h_{bPD} , h_{bLED}). Czulość kątowa standardowych fotodiod i efektywność emisji diod LED są opisywane funkcją kosinusa kąta padania / emisji światła przychodzącego / emitowanego pod kątem 90° do płaszczyzny elementu optycznego. Zastosowanie kartezjańskiego układu współrzędnych o początku zlokalizowanym pomiędzy dwoma środkowymi fotodiodami umożliwia opis położenia elementów sensora (Rys. 3.5). Pozycja każdego elementu określana jest przez środek powierzchni aktywnej elementu.



Rys. 3.5 Szkic centralnej części sensora (widok z boku) zawierający parametry opisowe.

Zdolność sensora do rozróżniania i obsługi zróżnicowanych gestów będzie wynikać z właściwości danej konfiguracji liniowego, optycznego sensora gestów. Zostały one opisane przez zbiór parametrów zdefiniowanych w kolejnych podrozdziałach. Ze względu na możliwość występowania w zależności od warunków otoczenia (np. od wystroju pomieszczenia) zróżnicowanej zawartości światła rozproszonego w świetle otoczenia czujnika gestów w stosunku do ilości światła kierunkowego, zdecydowano o analizie wartości

parametrów dla sensora pracującego w trybie aktywnym. W dalszej części badań przeprowadzona zostanie jednak ewaluacja właściwości czujnika pracującego również w trybie pasywnym.

3.3.1. Grupa kontrolna

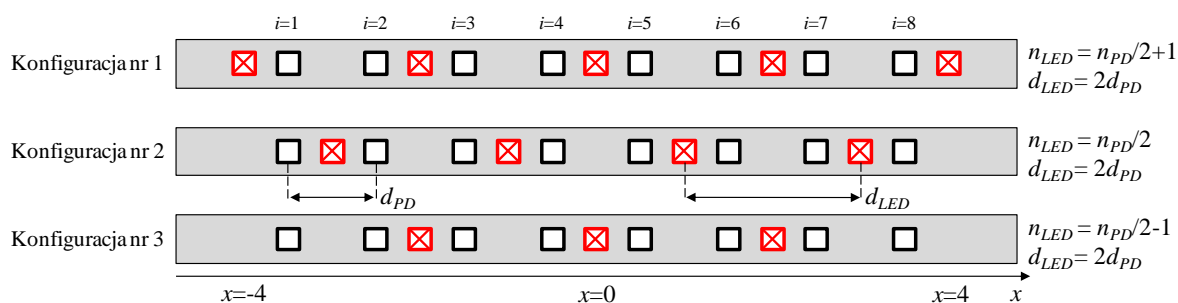
Wartości parametrów sensora w dużej mierze zależą od wymagań podyktowanych przez użytkowników systemu i ich charakterystykę. Celem zgromadzenia referencyjnych wielkości przeprowadzono badania na grupie kontrolnej składającej się z 41 osób rasy kaukaskiej (21 kobiet, 20 mężczyzn, średni wiek: $26,4 \pm 6,1$ roku).

W pierwszej części badań zmierzono wielkości fizyczne: promień palca wskazującego (R , poza 1FS), szerokość złączonych palców wskazującego i środkowego ($D2$, poza 2FJ) oraz odstęp między środkami rozpostartych palców środkowego i wskazującego (L , poza 2FS). Znajomość tych wielkości pozwoli na określenie takich właściwości sensora jak wymagana rozdzielczość przestrzenna i parametrów z nią związanych.

W drugiej części badań ochotników poproszono o wykonanie przesunięć dłonią wzdłuż osi x sensora, na wysokości od 1 do 5 cm nad jego powierzchnią. Każda osoba wykonała po 5 przesunięć danego typu: pozą 2FJ wolno, pozą 2FJ szybko oraz pozą 2FS szybko. Celowo nie podano wzorca prędkości dla szybkiego i wolnego przesunięcia aby każdy wykonywał je według własnego poczucia prędkości. Gesty rejestrowane były przez fotodiody sensora z częstotliwością 2kHz celem precyzyjnego wyznaczenia prędkości przesunięcia, przy włączonych diodach LED (tryb aktywny). Sygnały analogowe z fotodiod (elementy posiadały swój układ przetwornika prąd-napięcie) były bezpośrednio próbkowane przez zewnętrzny konwerter analogowo cyfrowy USB-1608GX DAQ. Eksperyment ten miał na celu oszacowanie wymaganej częstotliwości próbkowania sensora, dzięki której możliwe będzie zarejestrowanie przez czujnik zarówno wolnych jak i szybkich przesunięć dłonią.

3.3.2. Spłaszczenie profilu podświetlenia

Pierwszy z analizowanych w niniejszym rozdziale parametrów czujnika jest związany z miarą równomierności stopnia podświetlenia jego przestrzeni operacyjnej przez układ diod LED. Niech zależności między liczbą fotodiod i diod LED, odpowiednio n_{PD} i n_{LED} , oraz między d_{PD} a d_{LED} określają typ konfiguracji elementów optoelektronicznych w obrębie sensora gestów. W rozprawie doktorskiej zostały rozważone trzy typy konfiguracji (Rys. 3.6).



Rys. 3.6 Trzy typy konfiguracji elementów optoelektronicznych (widok z góry) w obrębie sensora liniowego. Bloki optyczne nie zostały zaprezentowane celem zwiększenia przejrzystości obrazu.

Ze względu na różnice geometryczne, konfiguracje tworzą swoje osobnicze profile intensywności stopnia podświetlenia przestrzeni przed sensorem, które mogą różnić się stopniem spłaszczenia wzdłuż osi x i y . Jednorodne natężenie światła pochodzącego od diod LED wzdłuż osi x na danej wysokości interpretowane jest jako maksymalne spłaszczenie. Im większe spłaszczenie, tym siła odbitego od dłoni światła mniej zależy od jej lokalizacji wzdłuż osi x , hipotetycznie umożliwiając wyznaczenie bardziej liniowej funkcji położenia wzdłużnego dłoni nad czujnikiem. Rozważmy płaską przeszkodę o szerokości w , położoną centralnie nad jedną z najbardziej wewnętrznych (pozycja 1) i nad jedną z najbardziej zewnętrznych fotodiod (pozycja 2), w odległości h od sensora, w dwóch osobnych pomiarach. Parametr $flatness_x$ jest obliczany jako stosunek siły sygnału uzyskanego w pozycji 2 do sygnału uzyskanego w pozycji 1 (dla przypadku $n_{PD}=8$, są to fotodiody o numerach 1 lub 8 i 4 lub 5). Parametr $flatness_y$ jest wyrażany jako funkcja $flatness_x(h)$. Im większy jest parametr $flatness_y$, tym większy efektywny zasięg sensora (fotodiody nasycają się dalej od urządzenia).

3.3.3. Strefy sensora

Kim i in. opisali swój dwudetektorowy sensor za pomocą trzech stref [51]. Obiekt interakcji (dłoń / palce wykonujące gest) znajduje się w *strefie martwej* (ang. *dead zone*) kiedy żaden z detektorów nie może go zobaczyć. W sensorze liniowym, rozważając przekrój wzdłużny, istnieje $n_{PD}+1$ takich stref, o trójkątnym kształcie (Rys. 3.7).

Jednak, kiedy czujnik pracuje w trybie aktywnym, przeszkoda musi znajdować się w polu widzenia zarówno detektorów jak i źródeł światła. Dlatego system podświetlenia posiada własną martwą strefę.

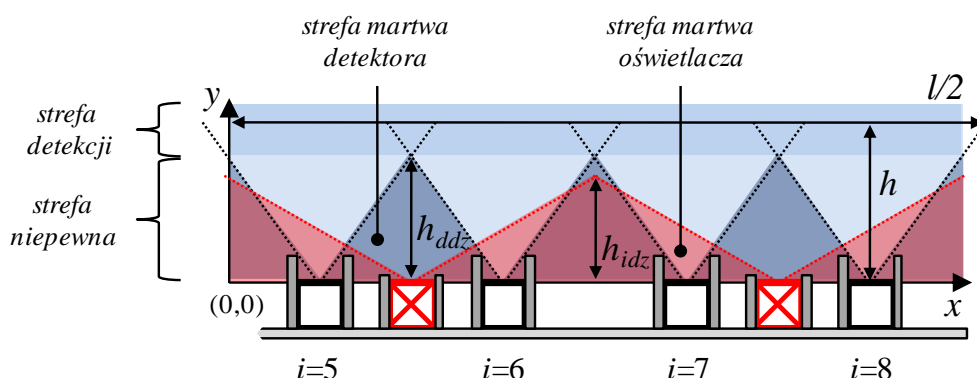
Wysokości poszczególnych martwych stref (detektora i oświetlacza / iluminatora) można opisać kolejno przez:

$$h_{ddz} = d_{PD} / (2 \operatorname{tg}(\beta_{PD} / 2)), \quad h_{idz} = d_{LED} / (2 \operatorname{tg}(\beta_{LED} / 2)). \quad (3.1)$$

Kiedy obiekt jest zlokalizowany bliżej, niż wynosi dystans operacyjny $h_{op} = \min(h_{ddz}, h_{idz})$, jego widoczność zależy od jego lokalizacji wzdłuż osi x . W tej strefie mogą zatem być otrzymywane niejednoznaczne wyniki, dlatego nazywa się ją *strefą niepewną* (ang. *ambiguous zone*) i powinna być ona minimalizowana.

Aby niezależnie od położenia wzdłuż osi x przeszkoda mogła być zawsze widoczna, musi znajdować się ona w *strefie detekcji* (ang. *detectable zone*), na wysokości $h \geq h_{op}$. Szerokość *strefy detekcji*, l (Rys. 3.7), na wysokości h , dla czujnika optycznego o elementach rozłożonych zbudowanego z n_{PD} fotodiod może być opisana przez:

$$l(h) = (n_{PD} - 1)d_{PD} + 2h \cdot \operatorname{tg}(\beta_{PD} / 2). \quad (3.2)$$



Rys. 3.7 Szkic prawej strony optycznego czujnika o liniowej budowie z zaznaczonymi trzema strefami.

Opisywany czujnik optyczny przeznaczony jest do detekcji gestów wykonywanych w polu bliskim w stosunku do czujnika, jako urządzenie do interakcji osobistej. Dlatego założono górny limit *strefy detekcji* gestu, $h_{max} = 5$ cm, jako że interferencja z obiektami położonymi w dalszej odległości nie jest pożądana. Ponieważ czujnik jest projektowany jako bezdotykowy, minimalny rozważany operacyjny dystans (tak aby można było swobodnie uniknąć dotknięcia) jest określany jako $h_{min} = 1$ cm, zatem $h_{op} \leq h_{min}$.

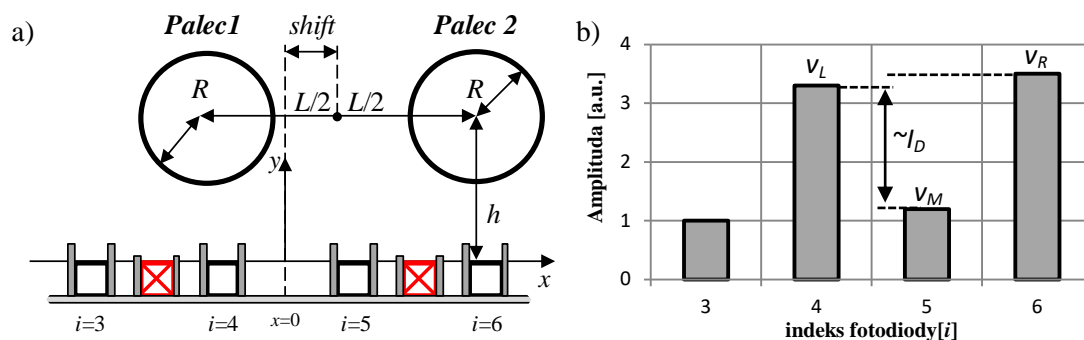
3.3.4. Zdolność rozdzielcza

Ważną właściwością sensora jest jego zdolność do rozróżniania konfiguracji palców typu FS i FJ czyli o palcach rozpostartych i złożonych (np. 2FS i 2FJ) co wymaga określonej zdolności rozdzielczej. Modyfikację zdolności rozdzielczej układu fotodiod w polu bliskim można zrealizować poprzez formowanie nadchodzącego promieniowania przez kolimatory, które ograniczają światło przychodzące pod ustalonym zakresem kątów. Obecność kolimatorów jest również niezbędna w tego typu konstrukcji ze względu na odcinanie przesłuchów światła z diod LED znajdujących się pomiędzy fotodiodami, a których światło bez takiego bloku wpada bezpośrednio przez obudowę fotodiody (niemal przezroczystą dla światła IR) na jej powierzchnię światłoczułą nasycając ją.

Rozważmy profil intensywności światła odbitego rejestrowany przez fotodiody, pochodzący od ułożenia palców w pozie 2FS. Przy wystarczającej zdolności rozdzielczej sensora zaobserwowane zostanie w profilu światła wcięcie. Ponieważ rozłożone palce będą odbijać światło diod LED, fotodiody położone bezpośrednio pod nimi odbiorą sygnał o wyższej wartości niż umieszczone np. pod przerwą między palcami. Dlatego przez dobór odpowiednich parametrów odległość h , na której czujnik wykryje opisywane wcięcie, powinna osiągać limit *strefy detekcji* czujnika, h_{max} . W konfiguracji 2FS palce o promieniu R są odseparowane od siebie na odległość L (Rys. 3.8a). Głębokość wcięcia, I_D (ang. *indentation depth*), w obserwowalnym profilu (Rys. 3.8b) jest kryterium separacji palców. Brak odpowiednio głębokiego wcięcia oznacza, iż wielkość L na jaką rozstawione są palce w konfiguracji 2FS, w odległości h od sensora, jest zbyt mała dla danego zestawu parametrów definiujących sensor. Z tego powodu, analizowany jest wpływ d_{PD} oraz β_{PD} na zdolność rozdzielczą sensora w dwóch osiach. Niech parametr *shift* oznacza dystans pomiędzy środkiem symetrii sensora ($x=0$) oraz położeniem składowej x -owej środka symetrii danego układu palców (Rys. 3.8a). Rozważana została zmienność parametru I_D w funkcji *shift*, h dla ustalonych R oraz L . Parametr I_D obliczany jest w sposób następujący:

$$I_D = 1 - \frac{v_M}{\min(v_L, v_R)}, \quad (3.3)$$

gdzie v_L i v_R oznaczają wartości szczytów otaczających minimum lokalne o wartości v_M , odpowiednio z lewej i prawej strony (Rys. 3.8b). Parametr I_D obliczany jest dla każdego minimum lokalnego w profilu natężenia światła, pod warunkiem, że zawiera on także co najmniej dwa maksima lokalne.



Rys. 3.8 a) Przekrój środkowej części sensora i palców w konfiguracji 2FS.
 b) Odpowiadający ułożeniu dłoni względem sensora ze schematu (a) profil intensywności światła rejestrowany przez fotodiody oraz oznaczenia istotnych wartości do wyznaczania głębokości wcięcia, I_D .

Pole widzenia elementów optycznych (β_{PD} , β_{LED}) może być w implementacji fizycznej sensora ustawiane za pomocą bloków optycznych o odpowiedniej wysokości (h_{bPD} , h_{bLED}). W trakcie badań ich wielkość dostosowywana była dzięki użyciu techniki druku 3D i umieszczaniu na sensorze wymiennych nakładek. Wydrukowana z przepuszczającego podczerwień materiału ABS nakładka wymagała nałożenia bezpośrednio na materiał kilku warstw metalizowanej farby akrylowej i dodatkowo na wierzch czarnej matowej farby akrylowej aby spełniała funkcję bloku optycznego. Obydwie farby naniesiono poprzez spray.

3.3.5. Czułość przestrzenna i czasowa

W literaturze detekcja ruchu (ang. *motion detection*) określa minimalny dystans (w płaszczyźnie xy), który musi pokonać ręka wykonująca gest aby zostało wykryte przesunięcie, w odległości h od czujnika [53]. W omawianym sensorze o budowie liniowej, czułość przestrzenna może być zdefiniowana jako minimalne zauważalne przez czujnik przemieszczenie obiektu poruszającego się wzdłuż osi x . Częstotliwość próbkowania czujnika i prędkość przemieszczenia się obiektu nie wpływają na tak zdefiniowaną czułość przestrzenną. Położenie obiektu znajdującego się przed sensorem, który tworzy profil intensywności światła odbitego, może być zatem ustalane na podstawie lokalizacji środka grawitacji, *COG* (ang. *Center Of Gravity*) takiego profilu (zakładając, iż pochodzi on od zwartej pozy dłoni):

$$COG = \frac{\sum_{i=1}^{n_{PD}} (v_i \cdot x_i)}{\sum_{i=1}^{n_{PD}} v_i}, \quad (3.4)$$

gdzie v_i to intensywność światła wyrażona w voltach, spróbkowana przez i -tą fotodiodeę zaś x_i to składowa wzdłużna pozycji tejże fotodiody w kartezjańskim układzie współrzędnych.

Celem jest analiza przebiegu rzeczywistej pozycji obiektu wzdłuż osi x w funkcji obliczonego COG . Rozważone zostaną przypadki obiektów różnego rodzaju (o różnej szerokości) i umieszczone w różnych odległościach od sensora.

Detekcja ruchu może być także rozważana w dziedzinie czasu. Częstotliwość próbkowania f_s sensora definiuje bowiem górny limit prędkości najszybszego notowalnego przemieszczenia dłoni, V_{not} . W takim wypadku mowa jest o czułości czasowej. Sensor gestów o liniowej budowie powinien dokonać co najmniej dwukrotnego próbkowania podczas przemieszczenia ręki wzdłuż sensora aby zauważyć przemieszczenie oraz oszacować prędkość tego ruchu. W najbardziej korzystnym przypadku warunku brzegowego, sensor pracujący z daną f_s wykona próbkowanie dokładnie w momentach kiedy ręka wykonująca ruch z prędkością V_{not} będzie znajdować się centralnie nad pierwszą i ostatnią fotodiodą szyku w następujących po sobie cyklach próbkowania. Podwojenie częstotliwości f_s zapewni minimum dwukrotne próbkowanie również w mniej korzystnych przypadkach synchronizacji momentów próbkowania i położeń dłoni. Z tego względu, minimalną częstotliwość próbkowania sensora liniowego, pozwalającą na wykrycie ruchu ręki nie przekraczającego prędkości V_{not} określa się za pomocą wzoru:

$$f_s = 2V_{not} / ((n_{PD} - 1) \cdot d_{PD}). \quad (3.5)$$

3.3.6. Pobór prądu

Parametry takie jak jasność diod LED, ich ilość, n_{LED} , docelowa f_s urządzenia oraz współczynnik wypełnienia, D , determinują poziom poboru prądu systemu podświetlającego sensora. Parametr D oznacza czas trwania okresu włączonych diod LED w stosunku do okresu próbkowania. Jego wartość jest ograniczona oddolnie przez czas ustalania sygnału układu wzmacniacza transimpedancyjnego oraz przez czas potrzebny konwerterowi ADC mikroprocesora na przeprowadzenie próbkowania. Z uwagi na to, że f_s wpływa na pobór prądu w trybie aktywnym (częstotliwość zapalania diod LED), wielkość ta powinna zostać zminimalizowana na podstawie wyników eksperymentu dotyczącego pomiaru prędkości przesunięcia dłoni przeprowadzonego z użyciem grupy ochotników (3.3.1).

3.4. Metodologia ewaluacji właściwości optycznego sensora gestów

Celem ewaluacji właściwości optycznego czujnika gestów o liniowej konstrukcji o elementach optoelektronicznych ułożonych w różnych konfiguracjach zdecydowano o dualnym charakterze prac badawczych. W efekcie, przeprowadzony został szereg badań zarówno eksperymentalnych jak i symulacyjnych. W trakcie eksperymentów wykorzystano między innymi fizyczną realizację testowej konfiguracji czujnika optycznego drugiej realizacji. Badania symulacyjne zostały przeprowadzone z wykorzystaniem zaprojektowanego i wykonanego środowiska interakcji światła z ciałami stałymi, zdolnego do przybliżenia właściwości fizycznego sensora. Wysoka zgodność symulacji i pomiarów pozwoli na przeprowadzenie części eksperymentów w środowisku symulacyjnym. Dzięki temu możliwa będzie weryfikacja testowej konfiguracji sensora wobec innych konfiguracji i wymagań oraz wykonanie wirtualnych eksperymentów trudnych do przeprowadzenia z odpowiednią kontrolą w świecie rzeczywistym i w znacznie krótszym czasie niż ten, który byłby potrzebny na ich fizyczną realizację (np. poprzez brak konieczności przygotowania nowych prototypów).

3.4.1. Stanowisko laboratoryjne

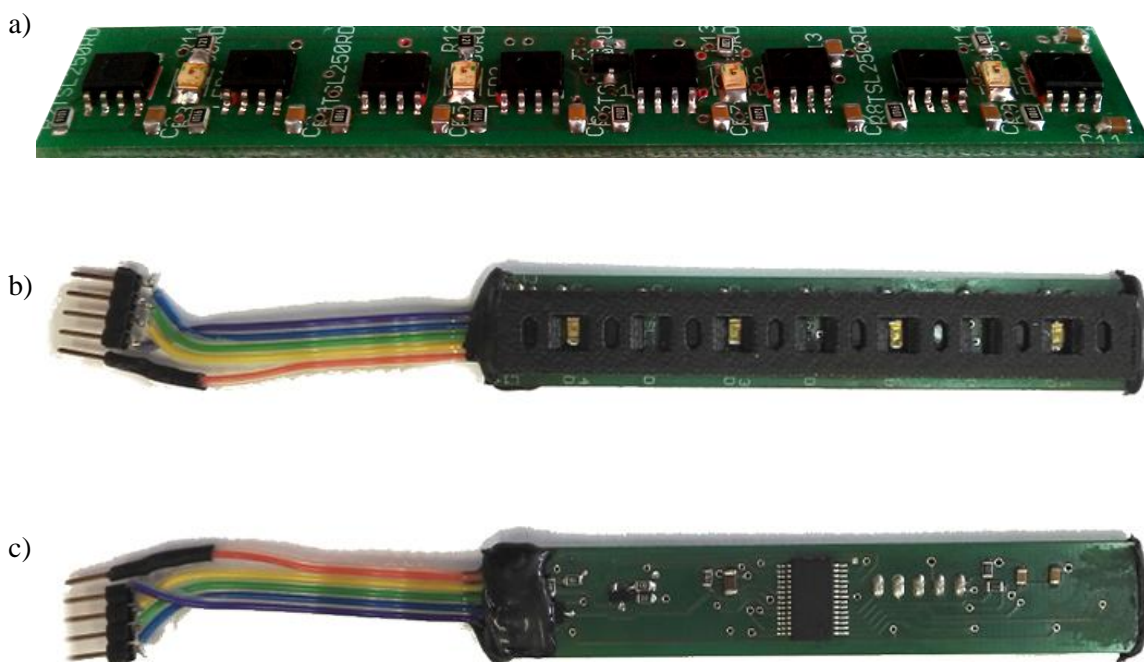
W skład stanowiska laboratoryjnego wchodziła: fizyczna realizacja optycznego sensora gestów drugiej realizacji, układ jezdny pozwalający na powtarzalne wykonywanie określonego rodzaju ruchów obiektów symulujących dłonie względem sensora oraz komputer PC odpowiadający za zapis danych z czujnika optycznego i programowanie układu jezdnego.

Testową konfigurację sensora drugiej realizacji, z użyciem której wykonywano pomiary zawarte w niniejszej rozprawie doktorskiej, stanowi konstrukcja z $n_{PD}=8$ elementami światłoczułymi TSL260RD oraz $n_{LED}=4$ diodami LED KA-3528SF4S. Elementy światłoczułe, dalej zwane w uproszczeniu fotodiodami, integrują fotodiode i wzmacniający układ transimpedancyjny zamieniający foto prąd na napięcie. Płytkę czujnika gestów została zaprojektowana pod zasilanie poziomem napięcia $V_{DD}=5$ V. Według noty katalogowej oraz według pomiarów, przy takim poziomie napięcia zasilania fotodiody TSL260RD nasycają się na poziomie ok. 3,8 V, co odpowiada irradiancji E_e ok. $65 \mu\text{W}/\text{cm}^2$ (dla światła o $\lambda=940$ nm, rezystora podciągającego wyjście do masy $R_L=10$ k Ω i temperatury otoczenia $T_A=25$ °C). Sygnały rejestrowane przez fotodiody są unipolarne i próbkowane przez blok ADC mikrokontrolera o 12 bitach, co odpowiada wartościom z zakresu 0-4095. Po przejściu przez blok ADC, na potrzeby analiz, wartość sygnałów

transformowane są z powrotem do dziedziny woltów (zakres 0-3,8 V). Ponieważ mikrokontroler zasilany jest poziomem 5 V, jest to jego domyślna górna wartość odniesienia przy konwersji analogowo cyfrowej (domyślna dolna wartość odniesienia to masa). Możliwe jest podanie do mikrokontrolera innej wartości odniesienia lecz w zaprojektowanej płytce PCB czujnika odpowiedni port został wykorzystany jako wejście analogowe jednej z fotodiod. Z tego względu należy zaznaczyć, iż stosowana konstrukcja pozwoliła na wykorzystanie ok. 76% użytecznego zakresu konwertera ADC. Większą dokładność konwersji przez podanie napięcia odniesienia można uzyskać poprzez zastosowanie mikrokontrolera z inną konfiguracją funkcjonalną i większą ilością portów. Schemat powstałej płytki PCB czujnika optycznego zamieszczono na końcu rozprawy doktorskiej w sekcji z materiałami dodatkowymi (Dodatek B Schemat PCB).

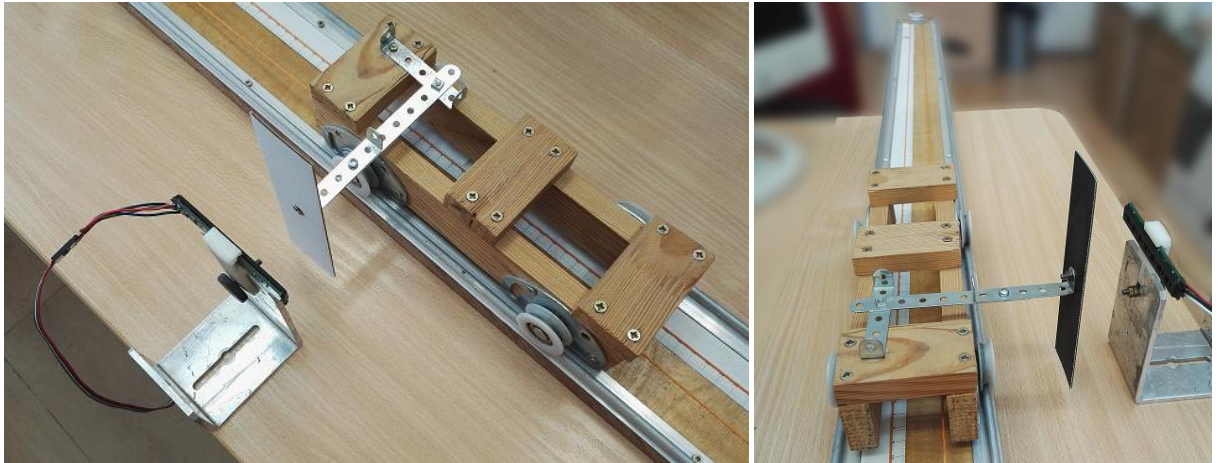
Pojedynczy okres próbkowania profilu natężenia światła generuje zatem n_{PD} wartości cyfrowych, które mogą być przeanalizowane przez mikrokontroler lub wysłane do innego urządzenia przez interfejs UART. Niech taki zestaw wartości (bez różnicy czy w postaci poziomów kwantyzacji czy po konwersji z powrotem na wolty) nazywa się dalej ramką danych (ang. *data frame*, DF).

Podawanie stanu wysokiego na odpowiednim wyjściu cyfrowym mikrokontrolera, podłączonego do układu klucza tranzystorowego, pozwala na sterowanie przepływem prądu przez gałęzie układu zawierające diody LED i tym samym ich włączanie. Diody LED podłączone są równolegle i każda posiada osobny rezystor (o tej samej wartości dla wszystkich LEDów) determinujący poziom jasności świecenia (przez wielkość prądu przezeń płynącego). W ustalonej konfiguracji, biorąc pod uwagę spadek napięcia na diodzie IR (1,3 V), poziom zasilania i wielkość rezystora (120 Ω) wynikowy prąd diody LED zapalanej w impulsie wynosił 30,83 mA. Wartości te (wartość opornika wpływająca na prąd) zostały dobrane eksperymentalnie. Czujnik używany był z zastosowaniem nakładki z blokami optycznymi zawężającej pole widzenia fotodiod. Konfiguracja geometryczna wykonanego czujnika opisana jest przez następujący zestaw parametrów: $d_{PD}=1$ cm, $\beta_{PD}=60^\circ$ i $d_{LED}=2$ cm, $\beta_{LED}=120^\circ$. Mikroprocesor znajduje się na dolnej stronie płytki PCB (Rys. 3.9).



Rys. 3.9 Fizyczna realizacja czujnika o eksperymentalnie dobranej konfiguracji elementów optoelektronicznych. Górna strona płytki bez zamontowanego bloku optycznego (a) oraz z blokiem (b) i dolna strona płytki (c).

W trakcie badań przeprowadzono szereg eksperymentów polegających na przemieszczaniu obiektów o różnym kształcie względem nieruchomo zamocowanego sensora. Do tego celu wykorzystano konstrukcję wózka poruszającego się po torze ruchem jednostajnym, prostoliniowym, który był napędzany silnikiem krokowym UHD-23-Z3 (Rys. 3.10). Sterowanie silnikiem odbywało się z poziomu osobnego układu mikrokontrolera i umożliwiało precyzyjne powtarzanie przejazdów wózka z zadaną prędkością i o ustalonym zakresie wychylenia w stosunku do pozycji sensora. W trakcie eksperymentów wózek z obiektami odbijającymi światło diod LED sensora (imitującymi pozy dłoni) poruszał się z prędkością 5 cm/s. Mikrokontroler czujnika gestów próbkował odbite światło z częstotliwością 40Hz i z rozdzielczością 12 bitów. Cyfrowe wartości natężenia światła odbitego były w każdym cyklu wysyłane przez port szeregowy UART do komputera i zapisywane na dysku do późniejszej analizy.



Rys. 3.10 Układ jezdny wraz z wagonikiem, do którego przyczepiono płaską przeszkodę. Układ może przemieszczać się naprzeciwko nieruchomo zamontowanego w specjalnym uchwycie optycznego czujnika o liniowej budowie.

3.4.2. Badania symulacyjne działania sensora optycznego

Celem przyspieszenia prac badawczych, przez sprawdzenie właściwości dowolnie skonfigurowanego czujnika optycznego bez konieczności budowy wielu prototypów, zdecydowano o utworzeniu środowiska symulacyjnego umożliwiającego przeprowadzenie szeregu wirtualnych eksperymentów. Model, na bazie którego takie środowisko mogłoby funkcjonować powinien uwzględniać interakcję światła z obiektami stałymi takimi jak wirtualne dłonie i bloki optyczne poprzez obliczanie wartości parametrów odbić fotonów, ich trajektorii oraz wypadkowego poziomu energii rejestrowanego przez fotodiody.

Zdecydowano o zastosowaniu modelu odbicia Lambertowskiego jako wyjściowego przybliżenia interakcji wirtualnych fotonów z wirtualnym obiektem dłoni. Potencjalnie wysoka zgodność symulacji i pomiarów, która zostanie zweryfikowana, upoważni autora rozprawy do poprzestania na Lambertowskim przybliżeniu modelu odbicia światła. W tym podstawowym modelu poziom natężenia światła odbitego zależy jedynie od energii fotonu przed zdarzeniem odbicia oraz kąta padania promienia światła i zmienia się wraz z kosinusem tego kąta. W badaniach wymagających dokładniejszego odwzorowania interakcji światła i skóry uwzględnia się także częściowe przenikanie światła przez skórę, zmienne właściwości różnych warstwy skóry czy długość fali światła. Modele takie opierają się często na funkcji fazowej Henyeya-Greensteina i opisywane są przez BRDF, które charakteryzuje właściwości refleksyjne powierzchni obiektu (ang. *Bidirectional Reflection Distribution Function*) [147][148].

Ze względu na złożoność mechanizmów stojących za modelowym odwzorowaniem źródła światła rozproszonego, zdecydowano o skupieniu się na symulacji trybu aktywnego

sensora gestów, kiedy dłonie wykonujące gest będą podświetlone przez diody LED (kilka źródeł punktowych). Ponadto, projektowane środowisko zostało uproszczone do dwuwymiarowego modelu opisanego w płaszczyźnie xy względem czujnika liniowego.

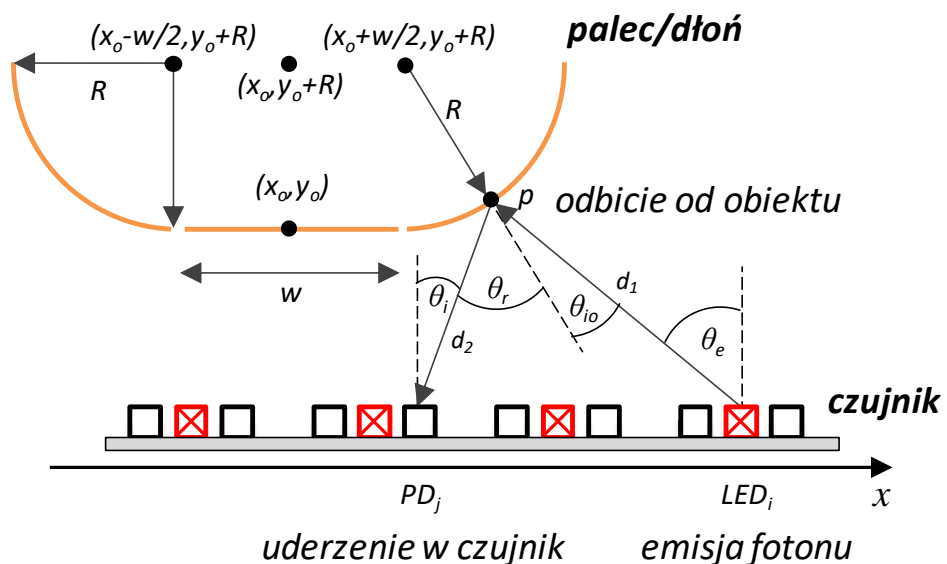
Zadaniem środowiska symulacyjnego jest możliwie wierne odwzorowanie *profilu natężenia światła* jaki zostałby zarejestrowany przez rzeczywisty czujnik o zadanej konfiguracji wobec konkretnie zlokalizowanego względem czujnika obiektu (dłoni / palców o uproszczonym kształcie). Celem uzyskania takiej charakterystyki często wykorzystuje się algorytmy oparte na metodzie Monte Carlo (MC). Polega ona na wielokrotnym losowaniu, zgodnie z odpowiednim dla zjawiska rozkładem, określonych parametrów cząstki, fotonu lub innego obiektu i statystycznym rozpatrzeniu jak sumarycznie zachowują się one wobec ograniczeń i zasad ujętych w obrębie zdefiniowanego modelu. MC wykorzystywane jest gdy analityczne rozwiązania stają się zbyt skomplikowane do sformułowania i obliczenia. Podejście to jest często stosowane w literaturze dotyczącej symulacji interakcji światła ze skórą niezależnie od stopnia złożoności modelu [148][149][150] czy renderingu obrazów [151][152]. Wobec zadania symulacji złożonego układu o rozłożonych źródłach światła (diody LED) oraz odbiornikach (fotodiod) a także zawierającego powierzchnie odbijające (palce dłoni) i pochłaniające (bloki optyczne) światło zdecydowano o projekcie symulatora czujnika optycznego bazującego na metodzie MC.

Projektowany model składa się z dwóch części: wirtualnych obiektów interakcji oraz algorytmu opisującego zachowanie fotonów. Na wirtualne obiekty składają się zasadniczo trzy typy elementów:

a) Modelowany czujnik, dowolnie konfigurowalny za pomocą wcześniej zdefiniowanych parametrów: n_{PD} , n_{LED} , d_{PD} , d_{LED} , β_{PD} , β_{LED} . Emituje fotony z diod LED i te z nich, które zostaną odpowiednio odbite od obiektu interakcji rejestruje za pomocą fotodiod. Diodę LED uproszczono do źródła punktowego podczas gdy powierzchnia aktywna fotodiody modelowana jest jako przedział (gdyż model jest dwuwymiarowy i opisuje przekrój wzdłuż osi xy) o długości $a=1$ mm (tak jak w elementach TSL260RD wykorzystanych w fizycznej implementacji czujnika). Model przewiduje wykorzystanie jednego czujnika w trakcie symulacji.

b) Modelowane bloki optyczne, za pomocą których ustalane jest pole widzenia elementów czujnika (β_{PD} , β_{LED}) dzięki ustawianiu ich wysokości oraz odległości od elementu (PD lub LED). W symulatorze ich właściwością jest, iż pochłaniają one fotony (a nie odbijają dalej) co w uproszczeniu ma odzwierciedlać interakcję w świecie rzeczywistym (fizyczna implementacja bloku jest matowa i czarna). Każdy element optoelektroniczny może posiadać dwa bloki optyczne (po jednym z każdej strony).

c) Modelowany obiekt interakcji, który może reprezentować jeden osobny palec lub 2 czy 4 złączone (odpowiednio pozy 1FS, 2FJ, 4FJ). Obiekt jest opisywany przez 4 parametry: x_o , y_o , R i w (Rys. 3.11). Punkt (x_o, y_o) niech określa pozycję środka czołowej strony obiektu względem czołowej strony czujnika. Parametr R reprezentuje promień zakrzywionej części obiektu podczas gdy parametr w opisuje szerokość części płaskiej (składowa wzdłużna) modelowanego palca/ów. Model umożliwia zatem zarówno uproszczenie pojedynczego palca do cylindra ($w=0$, $R>0$) i złączonych palców do powierzchni płaskiej ($w>0$, $R=0$), jak i wierniejsze ich odwzorowanie ($w>0$, $R>0$). Środowisko przewiduje symulacje z udziałem do 4 obiektów interakcji, co umożliwia przeprowadzenie badań także z udziałem póż dłoni o rozpostartych palcach (np. 2FS). W symulatorze ich właściwością jest, iż odbijają one fotony z efektywnością zależną od kąta padania i odbicia fotonu.



Rys. 3.11 Definicje geometrycznego modelu i parametrów symulatora.
Dla większej przejrzystości na schemacie nie zaprezentowano bloków optycznych.

W części algorytmicznej modelu definiowana jest całkowita liczba fotonów N_f biorąca udział w symulacji zatem każda z diod LED, zdefiniowanego według danej konfiguracji czujnika, emituje N/n_{LED} fotonów. W środowisku symulacji dla każdego źródła światła po kolei obliczane są trajektorie i energie sekwencyjnie emitowanych przez nie fotonów.

W fazie inicjalizacji fotonowi zostaje przypisana waga, W , równa jedności, która będzie maleć razem z długością przebytej przez niego drogi oraz efektywnościami różnych zdarzeń na jego drodze. Bezwymiarowa waga jest ekwiwalentem ilości energii fotonu. Obliczanie drogi fotonu w środowisku symulacji rozpoczyna wylosowanie kąta jego emisji, θ_e , przez i -tą diodę LED. Wartość ta determinuje efektywność emisji fotonu, η_e , opisywaną funkcją $\cos(\theta_e)$, która jest typową kątową charakterystyką promieniowania wielu diod LED. Następnie, jeśli foton emitowany pod kątem θ_e , na swojej drodze z punktu $(x_{LED_i}, 0)$ ominie bloki optyczne i napotka na obiekt interakcji, trafi go w punkcie p pod określonym kątem θ_{io} . Wtedy losowany zostaje kąt odbicia z tego punktu, θ_r , i obliczana jest uproszczona efektywność odbicia uwzględniająca obydwie kąty $\eta_{er} = \cos(\theta_{io} + \theta_r)$. Wartości obydwu stochastycznych kątów symulacji (θ_e i θ_r) losowane są z przedziału $(-\pi/2, \pi/2)$. W dalszej kolejności, jeżeli odbity od przeszkody foton ominie bloki optyczne, padnie na powierzchnię aktywną j -tej fotodiody pod kątem θ_i , jeśli dla $y=0$ składowa x -owa jego położenia znajdzie się w przedziale $\langle x_{PD_j-a/2}; x_{PD_j+a/2} \rangle$. Kąt ten wyznacza również efektywność pochłaniania fotonu przez fotodiode (η_i), której funkcję także można przybliżyć za pomocą funkcji kosinus. Po osiągnięciu tego etapu zostaje obliczona końcowa waga fotonu. Waga początkowa jest wymnażana przez obliczone efektywności emisji, odbicia i padania oraz dzielona przez kwadrat sumy całkowitej długości dwóch odcinków łamanej, z której powstała droga fotonu od źródła do detektora $(d_1 + d_2)$ (zgodnie z prawem odwrotnych kwadratów dla źródła punktowego):

$$I = \frac{I_0 \eta_e \eta_{er} \eta_i}{(d_1 + d_2)^2} \quad (3.6)$$

Wartości wag fotonów uderzających w powierzchnię czujnika w obrębie poszczególnych wirtualnych detektorów są na bieżąco sumowane przez każdą z fotodiod z osobna. Zgromadzone wielkości reprezentują intensywność światła mierzoną przez dany detektor. Otrzymany po wyemitowaniu N_f fotonów rozkład wartości odzwierciedla poszukiwany *profil natężenia światła* na powierzchni czujnika optycznego.

Listing 3.1 prezentuje pseudokod opisujący algorytm działania symulatora zaprojektowanego i zaimplementowanego przez autora niniejszej rozprawy.

```

for i_led = 1:n_led % dla każdego zdefiniowanego LEDa
    tmp_LED = get_LED(i_led) % pobierz obiekt danego LEDa (współrzędne)
    for i_photon = 1:Nf/n_led % emituj Nf/n_led fotonów z każdego źródła
        I0 = w % początkowa waga fotonu
         $\theta_e$  = sample_uniform(- $\pi/2$ ,  $\pi/2$ ) % losowanie kąta emisji
         $\eta_e$  = cos( $\theta_e$ ) % efektywność emisji

        % sprawdzenie czy foton trafia w obiekt
        hit_event_ob = get_hit_point(obstacle, tmp_LED,  $\theta_e$ )
        if(hit_event_ob.has_occurred) % jeśli nastąpiło uderzenie w obiekt
            % obliczenie kąta padania na przeszkodę
             $\theta_{io}$  = calc_incidence_angle(tmp_LED.point,
                                         hit_event_ob.point,
                                         obstacle)
             $\theta_r$  = sample_uniform(- $\pi/2$ ,  $\pi/2$ ) % losowanie kąta odbicia
             $\eta_{er}$  = cos( $\theta_{io} + \theta_r$ ) % efektywność odbicia

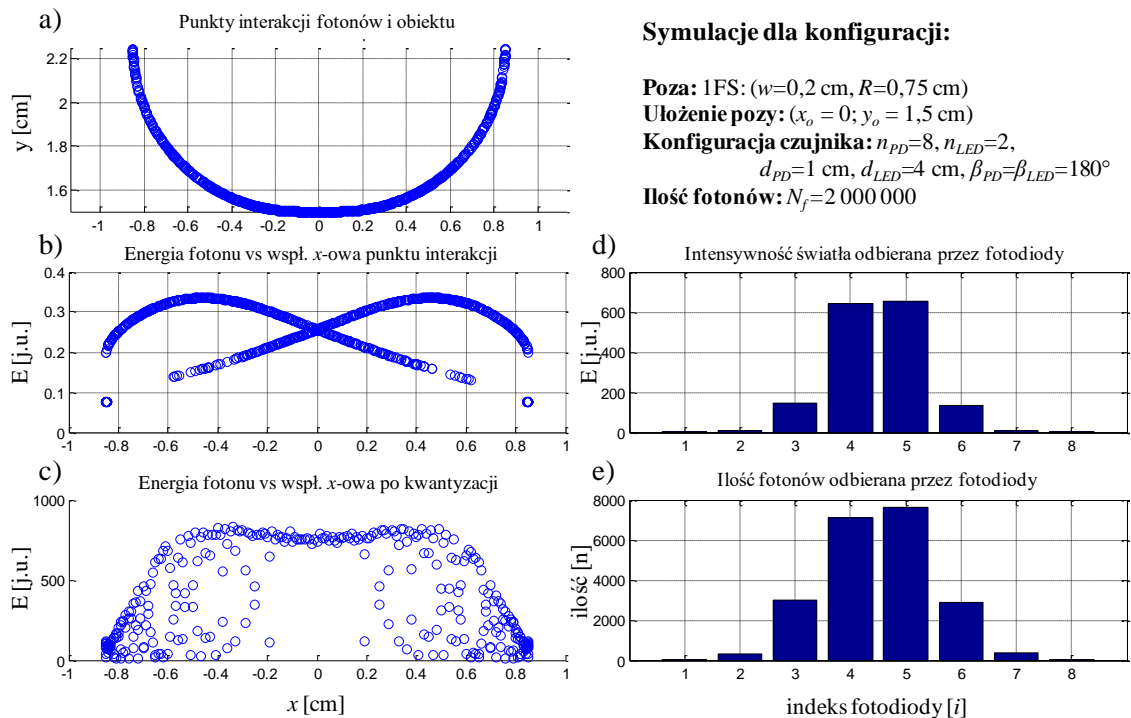
            % obliczanie trajektorii powrotnej fotonu
            hit_event_PD = get_hit_PD(PDs, obstacle,  $\theta_{er}$ )
            if(hit_event_PD.has_occurred) % jeśli trafił w fotodiode
                % obliczenie kąta padania na fotodiode
                 $\theta_{id}$  = calc_incidence_angle(hit_event_ob.point,
                                                hit_event_PD.point,
                                                sensor)
                 $\eta_{id}$  = cos( $\theta_{id}$ ) % efektywność pochłaniania fotonu
                % obliczanie trajektorii fotonu
                d1 = get_dist(tmp_LED.point, hit_event_ob.point)
                d2 = get_dist(hit_event_ob.point, hit_event_PD.point)
                % obliczanie końcowej wagi fotonu
                 $I = I0 \cdot \eta_e \cdot \eta_{er} \cdot \eta_{id} / (d_{obs\_det} + d_{LED\_obs})^2$ 
                % dodawanie energii fotonu do odpowiedniej fotodiody
                detector[hit_event_PD.PD_index] += I
    
```

Listing 3.1 Pseudokod symulatora interakcji światła z czujnikiem optycznym o liniowej budowie bazującego na metodzie Monte Carlo.

Walidacja algorytmu środowiska symulacyjnego oraz początkowe symulacje przeprowadzone zostały z wykorzystaniem środowiska MATLAB®. Duża liczba sekwencyjnie rozważanych fotonów (rzędu 10 milionów) wykazała jednak nieefektywność czasową tego środowiska do rozwiązań opartych na metodzie Monte Carlo. Model został zatem przeniesiony do języka C# i zoptymalizowany pod kątem wielowątkowości procesora. Pozwoliło to na skrócenie czasu działania symulatora ok. 75 razy używając tego samego komputera (procesor Intel G2130) i ok. 640 razy porównując algorytm działający w C#

na 7 wątkach na procesorze Intel i7-6700 w stosunku do algorytmu działającego w programie MATLAB® na jednym wątku na procesorze G2130.

Wynik przykładowej symulacji dla konfiguracji czujnika w postaci 8 fotodiod i 2 diod LED, dla obiektu imitującego pozę 1FS ($w=0,2$; $R=0,75$) umieszczonego w $(x_o=0, y_o=1,5)$, dla $N_f=2$ mln fotonów zamieszczono na Rys. 3.12.



Rys. 3.12. Wyniki symulacji MC dla obiektu interakcji imitującego pozę 1FS. a) Lokalizację punktów uderzenia fotonów w symulowany obiekt. b) Energia fotonów pochodzących z obydwu diod LED padających na powierzchnię palca (przed odbiciem) w funkcji wzdłużnej współrzędnej położenia (x). c) Energia fotonów na powierzchni palca po kwantyzacji (długość przedziału = 0,01 cm). d) Energia fotonów odbitych od obiektu odbierana przez fotodiody czujnika. e) liczba fotonów docierająca do każdej z fotodiod czujnika optycznego.

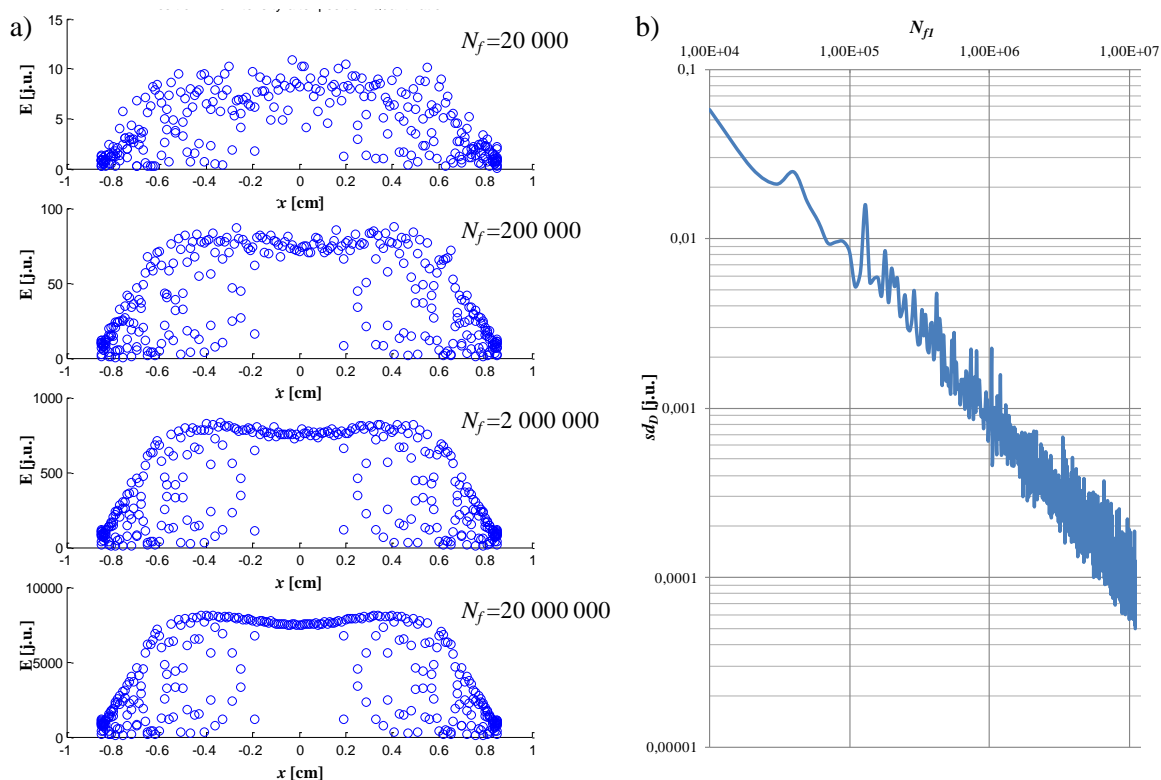
Symulacje typu Monte Carlo wiążą się z bardzo dużą liczbą losowań, w tym przypadku parametrów i trajektorii fotonów. Im więcej fotonów zostanie uwzględnionych w danej symulacji, tym będzie ona dokładniejsza. Niezależnie od stopnia przybliżenia z jakim implementowany model odzwierciedla rzeczywiste zjawisko należy odpowiedzieć na pytanie, jak wiele fotonów potrzeba wylosować, aby granulacja otrzymanego rozkładu bądź charakterystyki była na zadowalająco niskim poziomie. W niniejszym przypadku wynikiem każdej symulacji jest profil intensywności światła mierzony przez fotodiody, powstały przez odbicie fotonów od palców dłoni znajdujących się w polu bliskim czujnika. Dla małej ilości wyemitowanych fotonów (przykładowo 20 000) niewielka bezwzględnie liczba fotonów będzie odebrana przez fotodiody i uzyskany profil będzie zawierał znaczną ilość szumu. Na Rys. 3.13a zaprezentowano otrzymane profile

intensywności światła na powierzchni palca po kwantyzacji ($w=0,2$ cm, $R=0,75$ cm) znajdującego się w niewielkiej odległości od sensora ($x_o=0$ cm, $y_o=1$ cm) dla różnej liczby fotonów biorących udział w symulacji.

Liczbę fotonów wymaganą do przeprowadzenia symulacji Monte Carlo można oszacować za pomocą obiektywnego kryterium. Jak w przypadku metod iteracyjnych, może to być kryterium zbieżności mówiące o tym, czy kolejne obliczenia poprawiają dokładność wyniku w istotny sposób, z punktu widzenia zdefiniowanych wymagań. W trakcie formułowania kryterium wykorzystano fakt, iż poprawę dokładności danej symulacji wykonanej dla N_{f1} fotonów można osiągnąć poprzez dodanie cząstkowych obliczeń osobno wykonanej symulacji dla N_{f2} fotonów. Wystarczy jedynie dodać do siebie wektory z wynikami liczbowymi (np. profile intensywności światła) uwzględniając wkład liczbowy poszczególnych symulacji, czyli przemnożyć wyniki przez wagi $N_{f1}/(N_{f1}+N_{f2})$ oraz $N_{f2}/(N_{f1}+N_{f2})$. Uwzględnianie większej liczby fotonów w symulacji oprócz tego, że wygładza charakterystykę nie zmienia jej kształtu a jedynie mnoży przez stałą, związaną z liczbą fotonów w dodatkowej symulacji. Dlatego wyniki dwóch symulacji, przy założeniu niezmienności parametrów geometrycznych środowiska symulacji, dla N_{f1} i $N_{f1}+N_{f2}$ fotonów można porównać ze sobą poprzez normalizację (podzielenie przez maksimum globalne), odjęcie od siebie i obliczenie odchylenia standardowego z uzyskanego wektora różnic, sd_D . Kryterium można wobec tego sformułować następująco: zbieżność dla dwóch symulacji o liczbie fotonów N_{f1} i $N_{f1}+N_{f2}$ zostanie osiągnięta, gdy parametr opisujący stopień ich podobieństwa, sd_D , będzie mniejszy niż zadana wartość ε . W tym kryterium jedyną szukaną wartością jest N_{f1} . Wartość N_{f2} definiowana jest na potrzeby zautomatyzowania procesu odnalezienia szukanej wartości N_{f1} i od jej wielkości zależy dokładność wyznaczenia tego parametru. Jednak wobec spodziewanego rzędu wielkości N_{f1} wyrażanego w milionach, przyjęta wartość N_{f2} na poziomie 10 000 pozwoli na oszacowanie szukanego parametru z wystarczającą dokładnością. Informacja o rzędzie wielkości liczby fotonów potrzebnych do osiągnięcia zadanej dokładności obliczeń pozwala na optymalizację czasu symulacji a ustalenie N_{f2} na poziomie $\sim 0,01N_{f1}$ skraca jeszcze czas nadmiarowych obliczeń.

Dla czujnika o dużej ilości fotodiod ($n_{PD}=81$) i dla $N_{f2}=10\ 000$, dla pozy 1FS umieszczonej 1 cm centralnie nad czujnikiem przeprowadzono symulacje pod kątem oszacowania kryterium zbieżności dla wymaganej liczby fotonów (Rys. 3.13b).

Wynika z nich, iż dopiero dla około $N_{f1}=10\,000\,000$ osiągana jest zbieżność na poziomie $sd_D=1E-4$, którą, na podstawie obserwacji, uznano jako wartość kryterium ε . W dalszych symulacjach rozważane będą głównie czujniki o mniejszej liczbie detektorów (n_{PD} rzędu 8), odznaczające się tym samym niższymi wartościami sd_D . Jednak obiekty umieszczone dalej od czujnika powodują z kolei wzrost sd_D . Z tego powodu w dalszych symulacjach (o ile nie zostanie podana inna liczba) stosowana będzie liczba fotonów $N_f=20$ milionów.



Rys. 3.13 a) Energia fotonów na powierzchni pozy 1FS po kwantyzacji dla różnej ilości fotonów biorących udział w symulacji. b) Wykres sd_D pokazujących wpływ liczby fotonów biorących udział w symulacji na zbieżność uzyskiwanego wyniku.

3.4.3. Weryfikacja wyników symulacji względem pomiarów

Zaproponowano dwuetapową weryfikację zaprojektowanego modelu wirtualnego sensora względem pomiarów przeprowadzonych w świecie rzeczywistym dla eksperymentu skonfigurowanego analogicznie jak w badaniach symulacyjnych.

a) Ruch prostopadły obiektu do osi sensora

Pierwszy z etapów weryfikacji polega na pomiarze zależności amplitudy sygnału mierzonego przez poszczególne fotodiody od dystansu dzielącego sensor i obiekt odbijający światło LEDów. Przeprowadzono eksperyment, w którym płaski, biały karton został umieszczony naprzeciwko sensora i za pomocą układu jezdno został wprowadzony w ruch w ten sposób, iż zaczął oddalać się od niego. W efekcie, środek przeszkody, punkt P , przemieścił się między odległościami h_1 a h_2 (ścieżka prostopadła, Rys. 3.14a) przy składowej x -owej równej 0. Analogiczna próba została wykonana w symulatorze działania czujnika. Otrzymane drogą symulacji bezwymiarowe wartości z wirtualnych fotodiod należy skalibrować tak aby ich poziomy odpowiadały tym pomierzonym przez rzeczywisty sensor i wyrażonym w woltach. W tym eksperymencie rozważano uśrednioną amplitudę z dwóch środkowych fotodiod ($i=4, i=5$).

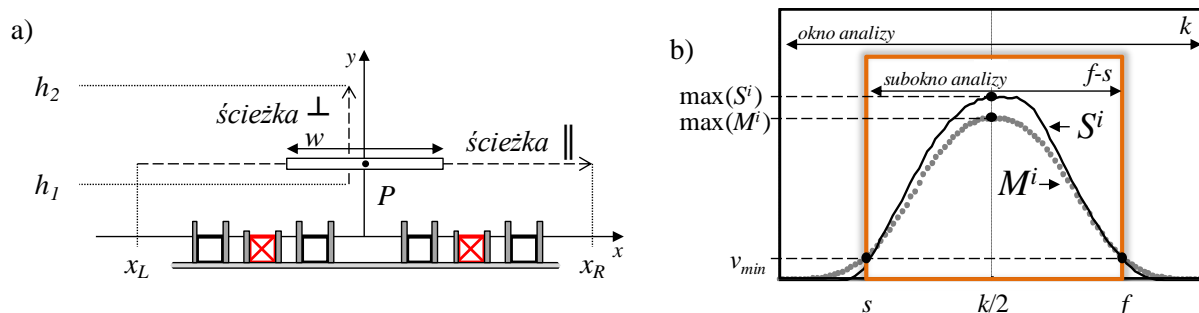
b) Ruch równoległy obiektu do osi sensora

Drugi etap weryfikacji ukierunkowany był głównie na analizę czułości kątowej wirtualnych fotodiod. W tym badaniu rozważono dwa różne typy przeszkód poruszających się wzdłuż sensora w konkretnej odległości h od niego tak, że środek przeszkody, P , poruszał się po prostej linii w zakresie wartości współrzędnej wzdłużnej od x_L do x_R (ścieżka równoległa, Rys. 3.14a) dla h o stałej wartości. Każdy taki przejazd obiektu wytwarza mierzony w czasie profil światła odbitego dla każdej z fotodiod. Niech S^i oraz M^i będą wektorami o długości k próbek przechowujących profil odbicia (zokienkowany tak, że środek okna został nałożony na próbkę profilu o maksymalnej wartości) z i -tej fotodiody, otrzymane odpowiednio drogą symulacji (S) i pomiaru (M). Założono, że długość profilu związanego z przejściem obiektu przed sensorem nie przekracza k próbek. Wektory S^i i M^i są normalizowane odpowiednio względem $\max(S)$ i $\max(M)$. Przebiegi uzyskane drogą symulacji i pomiaru, dla i -tej fotodiody będą porównywane za pomocą zaproponowanego parametru odległości:

$$\zeta^i = \sum_{j=s}^f |(S_j^i - M_j^i) / \max(M_j^i)| \cdot k / (f - s), \quad (3.7)$$

gdzie k oznacza długość okna analizy, s (start) oraz f (finisz) oznaczają indeksy próbek, które tworzą subokno analizy. Subokno zaczyna się/kończy w próbce o indeksie j , gdzie dowolny z przebiegów S^i, M^i osiąga/spada poniżej wartości v_{min} (Rys. 3.14b). Wartość v_{min} wynosi $2\% \cdot \max(M^i)$ i została dobrana eksperymentalnie. Zastosowanie subokna

do wzoru (3.6.) umożliwia miarodajne odniesienie do siebie wyników otrzymanych podczas eksperymentów z wąskimi i szerokimi obiektami przemieszczającymi się naprzeciwko sensora.



Rys. 3.14 a) Równoległa i prostopadła ścieżka obiektu względem sensora.
b) Widok zokienkowanych profili natężenia światła i koncepcja okna oraz subokna analizy (rysunek poglądowy).

3.4.4. Zgodność symulacji i pomiarów

a) Ruch prostopadły obiektu do osi sensora

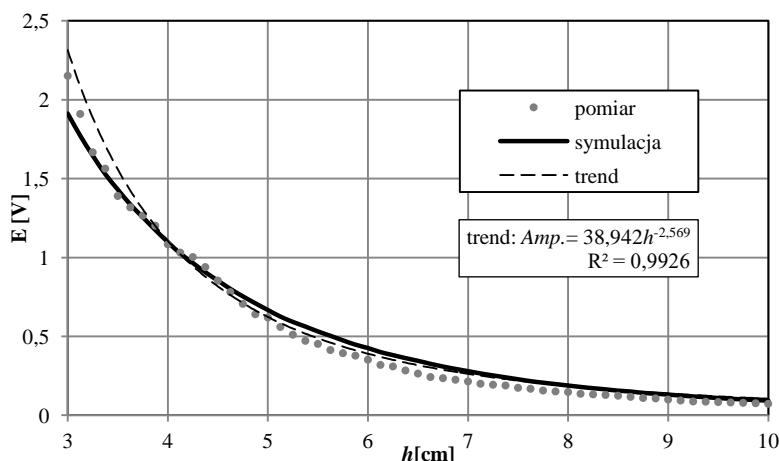
Zestawienie pomiarów i wyników symulacji z eksperymentu zaprojektowanego w ten sposób miało dwa cele. Pierwszym z nich była walidacja odpowiedzi wirtualnych fotodiod względem wartości otrzymanych z pomiarów przez porównanie podobieństwa ich kształtów. Drugim celem była kalibracja wielkości bezwymiarowego natężenia energii wirtualnych fotonów uzyskanych przez wirtualne detektory względem rzeczywistej wartości natężenia światła wyrażonej w woltach.

Walidacja symulacji w teście amplituda vs dystans została przeprowadzona z wykorzystaniem białej, płaskiej kartonowej przeszkody o szerokości $w=5$ cm (aby odbite światło oświetlało środkowe fotodiody możliwie jednolicie). Wartość h w tym eksperymencie i symulacji zmieniała się w zakresie od 3 do 10 cm. Celem porównania kształtów oraz kalibracji krzywej symulacyjnej S , względem pomiarowej M , zastosowano minimalizację miary rozbieżności RSS (ang. *residual sum of squares*, znaną również jako SSE, ang. *sum of squared errors of prediction*):

$$\hat{\omega}(Af) = \min_{Af} \sum_{i=1}^{Ns} (k_{Mi} - Af \cdot k_{Si})^2, \quad (3.8)$$

gdzie k_{Mi} oraz k_{Si} to odpowiednio i -te próbki krzywej pomiaru i symulacji uzyskane w tych samych odległościach h_i , Ns to ilość próbek, z których składają się krzywe, Af to współczynnik kalibracyjny. Za pomocą podanej formuły, minimalizowano błąd dopasowania krzywych ($\hat{\omega}(Af)$) poprzez dopasowywanie wartości współczynnika kalibracyjnego Af . Wartość znalezionej minimum wyniosła 0,48 dla współczynnika

kalibracyjnego A_f o wartości 29998,33. Wartość A_f jest liniowo zależna od liczby fotonów użytych w symulacji. Po kalibracji obydwa przebiegi zamieszczono na wspólnym wykresie (Rys. 3.15). Dodatkowo zaprezentowano funkcję trendu (wykres kropkowany), której formuła opisana jest przez prawo odwrotnej potęgi. Formuła ta jest bliska formule odwrotnych kwadratów, opisujących funkcję natężenia promieniowania od pojedynczego źródła punkowego w funkcji odległości od niego i potwierdza wiarygodność wyników otrzymanych dla układu wieloźródłowego.



Rys. 3.15 Zależność amplituda vs dystans (h) sygnałów z czujnika rzeczywistego i symulowanego.

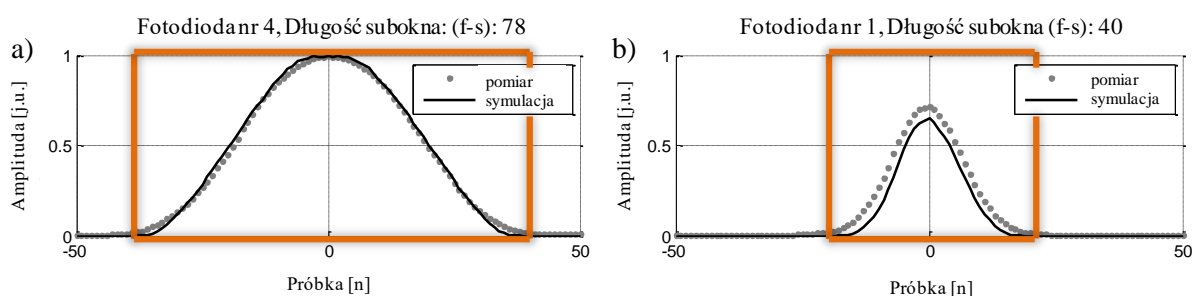
b) Ruch równoległy obiektu do osi sensora

W części eksperymentalnej testu na zgodność efektywnej czułości kątowej fotodiod (czułość kątowa elementów, wpływ bloków optycznych, odbicia od obiektów) dana przeszkoda dołączona do wagonika wykonała 20 przebiegów, podczas których mijała statycznie umieszczony sensor poruszając się w zakresie $x_L = -20$ cm do $x_R = 20$ cm (po 10 przebiegów z prawej do lewej i z lewej do prawej). Wzorcowy (najbardziej jednolity i gładki w swoim przebiegu) profil intensywności światła w czasie, wytworzony przez pojedynczy przejazd przeszkody wzdłuż sensora, został arbitralnie wybrany dla każdej i -tej fotodiody sensora. Następnie, spośród reszty profili wybrano 10 o najwyższej wartości współczynnika korelacji z profilem wzorcowym. Zostały one uśrednione tworząc wektory M^i . Celem zestawienia pomiarów (M^i) i symulacji (S^i) wykorzystano wzór (3.7). Pomiar i symulacje przeprowadzono dla dwóch różnych typów obiektów: białego płaskiego kartonika oraz białego cylindra, o parametrach, które opisuje Tabela 3.1. Tabela przedstawia także podobieństwo wyników według zaproponowanej miary odległości ζ (im mniejsza wartość tym większe podobieństwo).

Tabela 3.1 Odległość opisująca stopień podobieństwa symulacji (S) i pomiarów (M) dla dwóch typów obiektów.

fotodioda [i]	1	2	3	4	5	6	7	8
ζ^i (karton, $w=5$ cm, $h=4$ cm)	7,09	5,70	3,11	1,79	1,86	4,11	2,01	4,43
ζ^i (cylinder, $R=0,75$ cm, $h=2,8$ cm)	9,76	9,20	5,68	4,87	5,01	5,28	7,20	7,28

Rys. 3.16 pokazuje parę najlepiej dopasowanych profili (dla $i=4$) dla eksperymentu z płaskim kartonem (a) oraz najgorzej dopasowaną parę pod względem parametru ζ dla eksperymentu z cylindrem (dla $i=1$) (b). Ramka na wykresie oznacza długość subokna.



Rys. 3.16 Porównanie kształtów otrzymanych przy symulacji i pomiarów dla okna o długości k próbek.

Uznano, iż poziom zgodności wyników uzyskanych na drodze pomiarów eksperymentalnych i badań symulacyjnych pozwala na przeprowadzenie dalszych analiz właściwości czujnika optycznego o budowie liniowej z wykorzystaniem zaprojektowanego środowiska symulacyjnego.

3.5. Wyniki pomiarów i symulacji

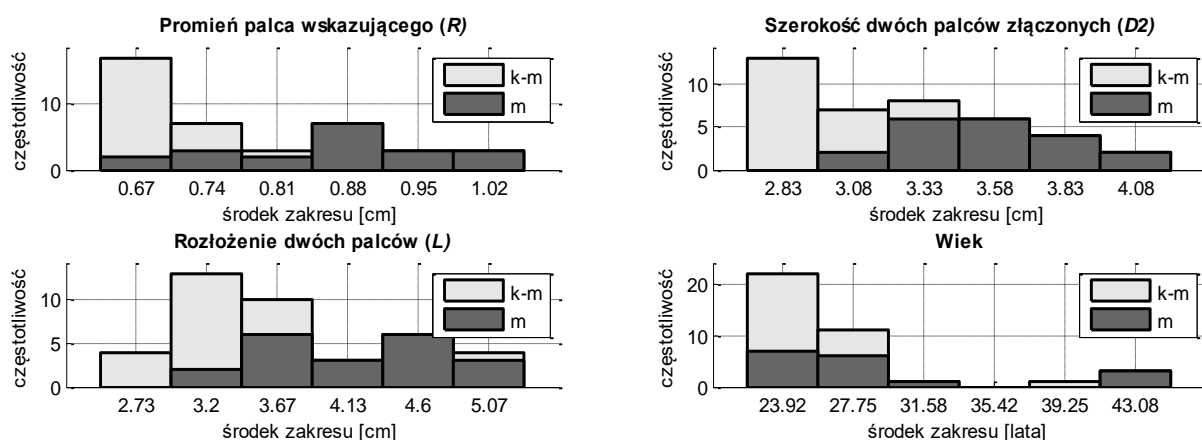
3.5.1. Grupa kontrolna

Wartości uzyskane po obliczeniu statystyki na wynikach z grupy kontrolnej zostały wykorzystane jako wartości referencyjne przy definiowaniu wymagań, obliczaniu parametrów technicznych i wykonywaniu symulacji dla optycznego czujnika gestów. Tabela 3.2 zawiera podsumowanie badań przeprowadzonych na 41 ochotnikach.

Tabela 3.2 Statystyczne parametry grupy kontrolnej. k – kobiety, m – mężczyźni.

Parametr	Mediana kobiety	Mediana mężczyźni	Mediana k+m	Średnia k+m	Odch. std. kobiety	Odch. std. mężczyźni	Odch. std. k+m
R [cm]	0,7	0,85	0,75	0,78	0,05	0,10	0,11
$D2$ [cm]	2,9	3,6	3,2	3,26	0,18	0,30	0,41
L [cm]	3,1	4,0	3,6	36,9	0,52	0,65	0,76
wiek [lata]	23	25,5	24	26,41	3,72	7,64	6,11

Wyniki przedstawiono również w formie histogramów z rozróżnieniem na płeć ochotników (Rys. 3.17).

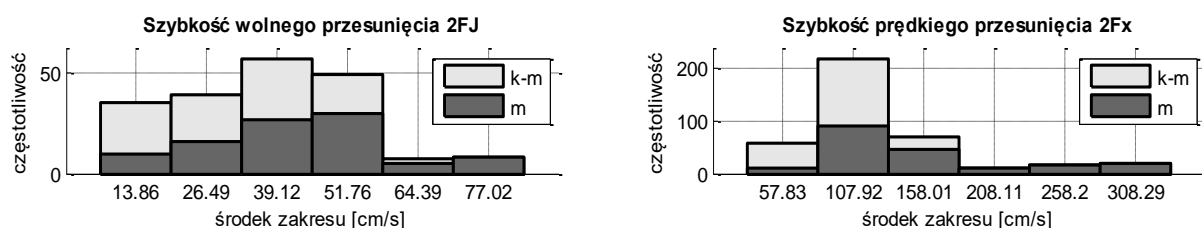


Rys. 3.17 Histogramy wielkości pomierzonych i osobniczych grupy ochotników. k – kobiety, m – mężczyźni.

Z uwagi na to, że poczucie prędkości jest cechą mocno osobniczą, wyniki uzyskane z przeprowadzonych pomiarów były zróżnicowane. Aby prezentowane histogramy (Rys. 3.18) były odpowiednio zwarte, na potrzeby utworzenia wykresu, odrzucono wartości leżące w odległości większej niż trzy odchylenia standardowe od średniej dla ruchów szybkich i wolnych (lecz uwzględnia je Tabela 3.3).

Tabela 3.3 Statystyka gestów przesunięcia dłonią wykonanych przez grupę ochotników. k – kobiety, m – mężczyźni.

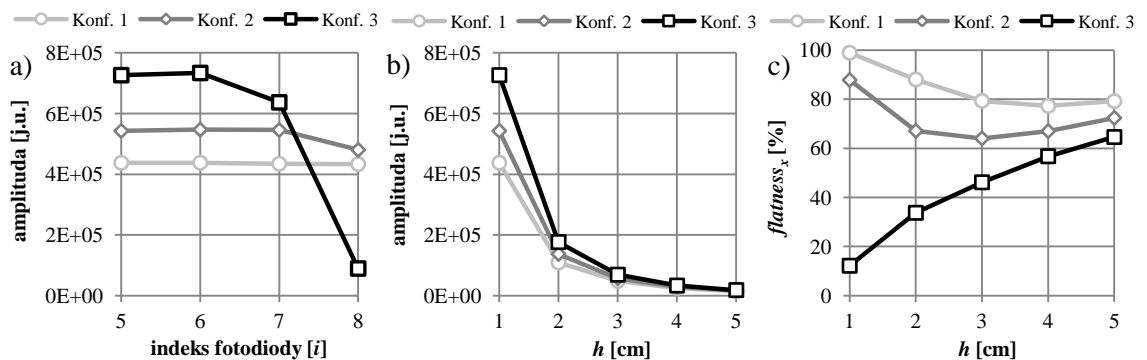
Gest	Mediana kobiety	Mediana mężczyźni	Mediana k+m	Średnia k+m	Odch. Std. kobiety	Odch. Std. mężczyźni	Odch. Std. k+m
2FJ wolno [cm/s]	33,90	43,02	37,04	39,38	15,85	20,49	19,08
2Fx=2FJ/2FS szybko [cm/s]	95,24	125	105,26	132,68	37,42	110,53	87,47



Rys. 3.18 Histogramy ruchów przesunięć dłonią wykonanych przez grupę ochotników. k – kobiety, m – mężczyźni.

3.5.2. Spłaszczenie profilu podświetlenia

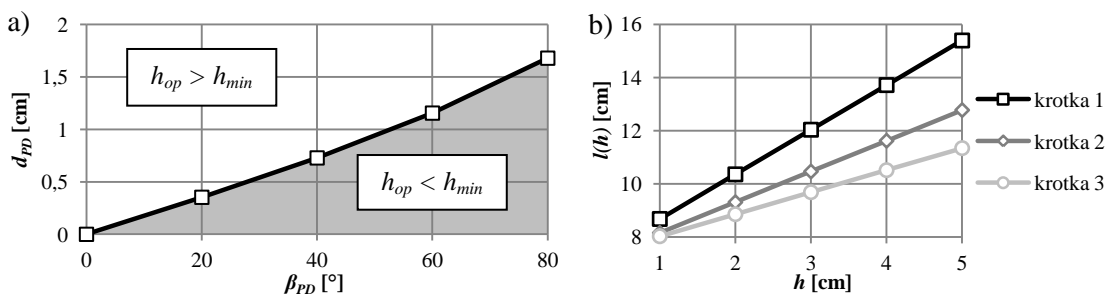
Przy symulacjach każdej z rozważanych konfiguracji sensora wyemitowana została ta sama liczba fotonów, N_f (50 milionów). Możliwe jest zatem bezpośrednie porównanie efektywności systemów iluminacji (np. w funkcji odległości) ponieważ dzięki temu wirtualnie rozproszyły one tę samą ilość energii. Rozważany w symulacji płaski obiekt imitował pozę 2FJ ($w=3,2$ cm - mediana parametru D_2 , Tabela 3.2). W celu zawężenia ilości rozważanych zmiennych problem spłaszczenia wzoru podświetlenia rozważono przy stałych wartościach $\beta_{PD}=60^\circ$, $\beta_{LED}=120^\circ$, $n_{PD}=8$ i $d_{PD}=1$ cm lecz dla wszystkich trzech rozważanych konfiguracji (o wartości n_{LED} odpowiednio równej 5, 4 i 3 dla Konfiguracji nr 1, 2 i 3). Wartości sygnałów z poszczególnych fotodiod dla obiektu znajdującego się w odległości h tworzą profile podświetlenia, np. dla $h=1$ cm (Rys. 3.19a). Efektywność systemu podświetlenia została rozważona dla środkowych fotodiod (Rys. 3.19b). Siła sygnału w odległości $h \geq 3$ cm jest zbliżona dla wszystkich konfiguracji lecz dla $h=1$ cm widoczne jest znaczne jej zróżnicowanie. Należy zadbać aby jednocześnie siła sygnału nie była mała w polu dalekim (aby sensor był w stanie doświetlić obiekty znajdujące się w *strefie detekcji*) i zbyt duża w polu bliskim (aby poprzez nasycenie się fotodiod nie tracić informacji o kształcie profilu natężenia światła w zbyt dużej odległości od czujnika). Wartości $flatness_x$ wskazują na poziom spłaszczenia profilu podświetlenia wzdłuż osi x na danej wysokości nad sensorem. Funkcje $flatness_x(h)$, oznaczane też jako $flatness_y$ (Rys. 3.19c), pozwalają na oszacowanie jednorodności profilu oświetlenia w płaszczyźnie xy .



Rys. 3.19 a) Siła sygnału światła odbitego od obiektu imitującego pozę 2FJ, umieszczonego nad i -tą fotodiodą na wysokości $h=1$ cm. b) Siła sygnału światła odbitego z $i=5$ fotodiody w funkcji h . c) Funkcje $flatness_x$ utworzone z poszczególnych wartości $flatness_x$.

3.5.3. Strefy sensora

Wysokość strefy niepewnej, h_{op} , powinna być utrzymywana poniżej h_{min} . Relacja ta jest spełniona dla $d_{LED}=2d_{PD}$, $\beta_{LED}=120^\circ$ i wartości β_{PD} leżącej w zakresie do 82° (wzór 3.1). Relacja d_{PD} vs β_{PD} , wynikająca ze wzoru 3.1, pomaga wypełnić warunek $h_{op} \leq h_{min}$ (Rys. 3.20a, powierzchnia zamalowana). Niech w niniejszej rozprawie krotka opisuje zestaw trzech parametrów: n_{PD} , d_{PD} , β_{PD} . Wartość $l(h)$ była obliczana dla różnie skonfigurowanych czujników, opisanych przez 3 krotki: (6; 1,4; 80°), (8; 1; 60°) i (10; 0,8; 45°) (Rys. 3.20b). Wartości n_{PD} i d_{PD} z każdej krotki były dobierane tak aby iloczyn $(n_{PD}-1) \cdot d_{PD}$, które oznaczają odstęp między zewnętrznymi fotodiodami, $l(0)$, były możliwie równe. Ich wielkość została dobrana tak, aby była większa od standardowej szerokości dłoni uzyskanej z pomiarów na grupie ochotników ($2 \cdot D_2=6,4$ cm) by sensor miał szansę postrzegać całą dłoń umieszczoną centralnie naprzeciwko niego. Parametry d_{PD} i β_{PD} z rozważonych krotek pozwalają na spełnianie przez czujniki warunku $h_{op} \leq h_{min}$. Mniejsze wartości d_{PD} nie są rozważane z powodu rozmiaru obudowy zastosowanych fotodiod (5 mm), ze względu na konieczność zmieszczenia bloków optycznych i diody LED w przestrzeni pomiędzy fotodiodami.

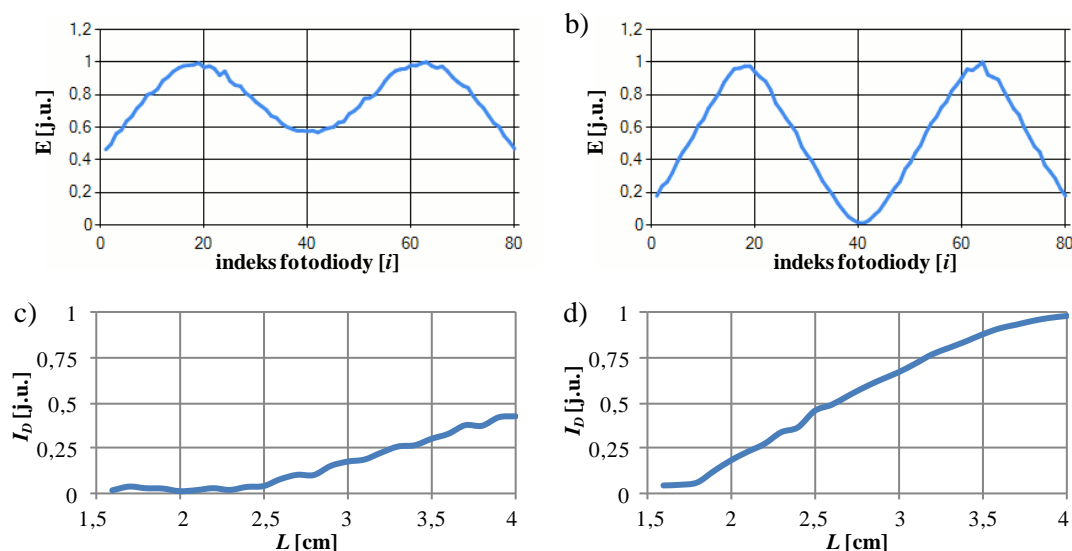


Rys. 3.20 a) Funkcja łącząca wartości β_{PD} i d_{PD} pozwalające na utrzymanie h_{op} na właściwym poziomie (strefa zamalowana). b) Parametr $l(h)$ dla trzech różnych konfiguracji sensora opisanych przez zdefiniowane krotki.

3.5.4. Zdolność rozdzielcza

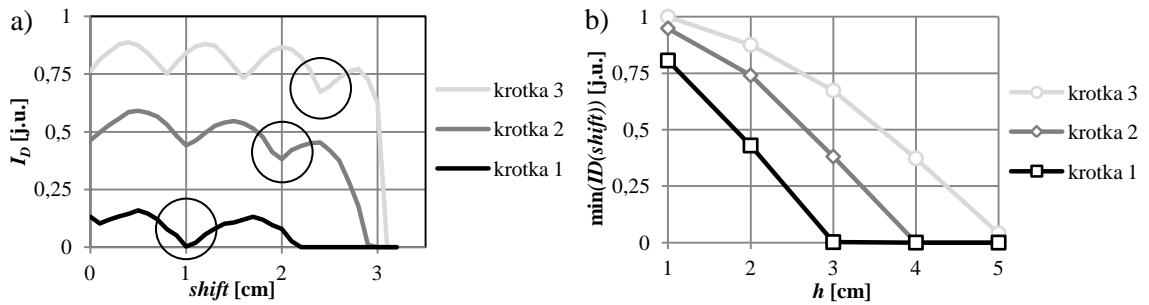
Celem demonstracji korzystnego wpływu stosowania bloków optycznych na zdolność rozdzielczą czujnika (wzrost wartości parametru I_D) zasymulowano obecność poży 2FS ($w=0$, $R=0,75$ cm, Tabela 3.2) o początkowym rozstawie palców $L=4$ cm, umieszczonej na wysokości $h=3$ cm dla $x_o=0$ cm. Dla wirtualnego czujnika o dużej liczbie detektorów ($n_{PD}=81$, aby precyzyjnie zaobserwować zmienność I_D) i jednorodnie promieniującej powierzchni (rozproszone źródła światła) rozpatrzono dwa eksperymenty: wpływ braku bloku optycznego ($\beta_{PD}=180^\circ$) oraz jego obecności ($\beta_{PD}=60^\circ$) na głębokość wcięcia w profilu intensywności światła odbitego. Zmienną w symulacji była wartość L

malejąca z krokiem 0,1 cm do 1,5 cm czyli aż do złączenia palców ($R+R=1,5$ cm)(Rys. 3.21). Zaprezentowane wykresy (Rys. 3.21c, d) wskazują, że dla wartości progowej $I_D=0,15$ tolerancja czujnika na rozróżnianie dwóch palców na wysokości $h=3$ cm jest większa o 1 cm dla czujnika z blokami optycznymi niż bez (2 cm vs 3 cm).



Rys. 3.21 Symulacje wpływu bloków optycznych na zdolność rozdzielczą czujnika optycznego. Widoczność pozy 2FS o $L=4$ cm przez czujnik bez bloków optycznych, $\beta_{PD}=180^\circ$ (a) oraz z zastosowaniem bloków, $\beta_{PD}=60^\circ$ (b). Przebieg parametru I_D w funkcji malejącego odstepu między palcami, L dla braku bloków optycznych (c) i przy obecności (d).

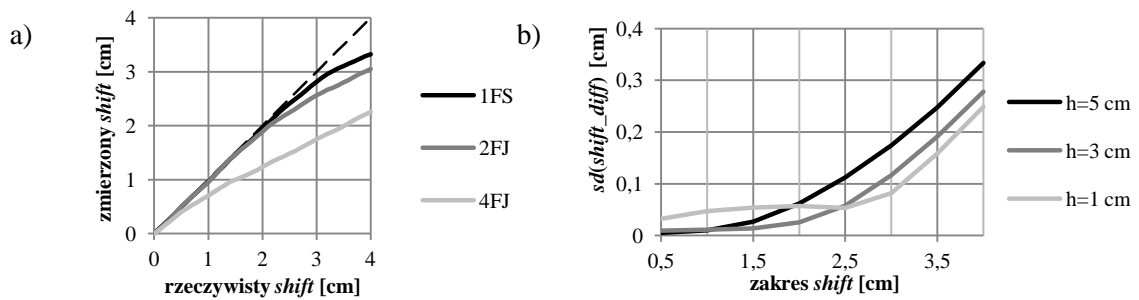
Następnie, również w symulatorze, rozważone zostały wirtualne czujniki opisane przez 3 krotki, skonfigurowane tak jak w poprzedniej sekcji ale z wartością $\beta_{LED}=120^\circ$, a każdy z nich z ilością elementów i ułożeniem jak w Konfiguracji nr 1. Wartości parametrów pozy 2FS: R i L zostały wzięte z odpowiednich median z badania przeprowadzonego na grupie ochotników (Tabela 3.2). Na Rys. 3.22a zaprezentowano funkcje $I_D(shift)$ dla referencyjnej wysokości około połowy zasięgu czujnika ($h=3$ cm). Widoczne są na nich fluktuacje (w odróżnieniu od sytuacji z Rys. 3.21), których przyczyną jest mniejsza ilość detektorów i źródeł światła w tym eksperymencie. Najmniejsze z lokalnych minimów przebiegów (wcięcie pomiędzy sąsiednimi wybrzuszeniami funkcji I_D), które zaznaczono okręgami, są rozważane jako minimalne I_D na danej wysokości h , podczas wykreślania funkcji $\min(I_D(shift))$ vs h (Rys. 3.22b).



Rys. 3.22 a) Funkcje $I_D(\text{shift})$ obliczone dla standardowej pozy 2FS umieszczonej na wysokości $h=3$ cm, dla każdej z trzech krotek sensora. b) Funkcje $\min(I_D(\text{shift}))$ vs h dla trzech krotek.

3.5.5. Czułość przestrzenna i czasowa

Trzy typy obiektów imitujących zwarte pozy odbijające światło zostały rozważone w symulacjach dotyczących czułości przestrzennej sensora. Jako pozę 1FS zasymulowano walec ($R=0,85$ cm) zaś jako pozy 2FJ i 4FJ zasymulowano płaszczyzny odpowiednio o $w=3,2$ cm (mediana $D2$, Tabela 3.2) oraz o $w=6,4$ cm (podwojona wartość $D2$). Na początku każdej z symulacji dany obiekt umieszczany był w wirtualnej przestrzeni tak, aby środek jego strony czołowej (punkt (x_0, y_0)) znajdował się w punkcie $(0, h)$ - położenie centralnie nad czujnikiem. Następnie, obiekt był przemieszczany wzdłuż osi x co powodowało, że wielkość parametru shift ulegała zmianie. Wirtualne próbkowanie wykonywane było co zmianę parametru shift o $0,1$ cm, zatem rzeczywista wartość shift (położenie sensora) była znana w każdym kroku symulacji. Położenie zmierzone (shift zmierzony), które zaobserwuje sensor obliczyć można na podstawie wzoru na środek grawitacji profilu natężenia światła, COG (wzór 3.4). Obliczone zostało ono dla każdego z trzech obiektów umieszczonych na wysokości $h=3$ cm, rozważając sensor skonfigurowany jak w krotce 3 (sekcja 3.3.4 Zdolność rozdzielcza), o rozłożeniu elementów jak w Konfiguracji nr 1 (Rys. 3.23a). Odchylenie standardowe różnicy pozycji rzeczywistej i zmierzonej, $sd(\text{shift_diff})$, na różnych wysokościach h od sensora zostało rozważone dla obiektu imitującego pozę 2FJ. Błędy obliczone zostały dla różnic o rosnącym co $0,5$ cm zakresie, kolejno: $0-0,5$ cm, $0-1$ cm aż do $0-4$ cm (Rys. 3.23b). Jako referencyjną wybrano pozę 2FJ ponieważ odbija ona więcej światła niż poza 1FS (jest zatem lepiej widoczna w większej odległości) oraz wykazuje się większą precyzją odwzorowania położenia niż 4FJ (mniejsze nieliniowości) (Rys. 3.23a).



Rys. 3.23 a) Zmierzony vs rzeczywisty shift dla różnych pól umieszczonych na wysokości $h=3$ cm.
 b) Odchylenie standardowe różnicy pozycji zmierzonej i rzeczywistej wzdłuż osi x dla pozycji 2FJ w funkcji h .

Tabela 3.4 pokazuje zależności między częstotliwością próbkowania czujnika f_s a notowaną prędkością przesunięcia dłonią, V_{not} , dla sensora skonfigurowanego tak jak opisuje krotka 2 (czyli takiego jak w fizycznej implementacji sensora drugiej realizacji). Ze względu na możliwie równe iloczyny $(n_{PD}-1) \cdot d_{PD}$ wartości estymowanych prędkości i częstotliwości będą takie same lub bardzo zbliżone dla sensorów skonfigurowanych także jak w pozostałych dwóch krotkach. Tabela 3.4 zawiera opisy kilku szczególnych przypadków, *Pr*. *Pr6* pokazuje wartość V_{not} dla najwyższej osiągalnej f_s dla której możliwe jest spróbkowanie wartości z 8 kanałów analogowych, biorąc pod uwagę także czas ustalania sygnału na fotodiodach. W *Pr1* i *Pr5* V_{not} wybrano odpowiednio jako medianę i wartość maksymalną z szybkich przesunięć dłonią wykonanych przez grupę kontrolną. *Pr2-4* pokazują najszybszą notowaną prędkość przesunięcia dla trzech wybranych (okrągłych) częstotliwości próbkowania. Tabela pokazuje, jak wartość f_s wpływa na D a zatem i na pobór prądu przez czujnik. Wartość parametru C wskazuje jaka część szybkich przesunięć dłonią mogłaby zostać wykryta przy zastosowanej częstotliwości próbkowania. Dlatego wybór f_s powinien być kompromisem między parametrami C a D .

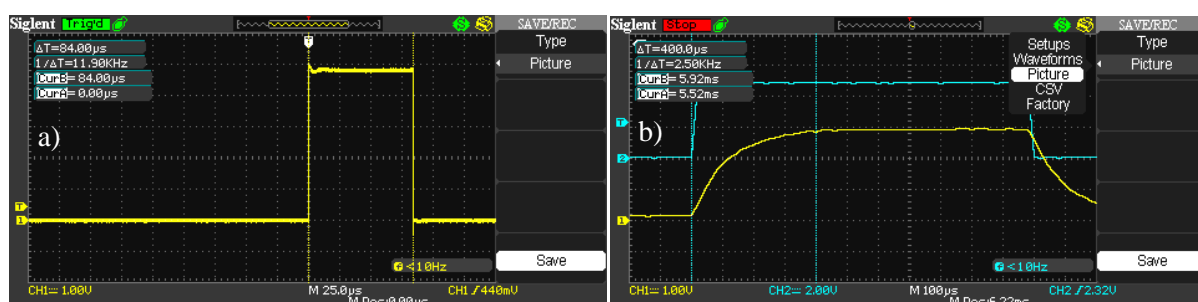
Tabela 3.4 Wpływ częstotliwości próbkowania na efektywność energetyczną i funkcjonalną sensora gestów.

Pr	Dana f_s [Hz]	V_{not} [cm/s]	Dana prędkość [cm/s]	Wymagana f_s [Hz]	C [%]	D [%]
1	-	-	105,26	29,24	53,4	1,13
2	40	140	-	-	74,9	1,50
3	50	175	-	-	84,2	1,88
4	100	350	-	-	98,3	3,75
5	-	-	666,67	185,19	100	7,14
6	2666,67	9333,33	-	-	100	100

3.5.6. Pobór prądu i kalibracja

Wobec zaproponowanego schematu elektrycznego sensora gestów o budowie liniowej (Dodatek B Schemat PCB) minimalizacja poboru prądu mogła być rozważona głównie przy czasie zapalania diod LED i czasie próbkowania. W celu wykonania pomiarów referencyjnych posłużono się oscyloskopem cyfrowym Siglent SDS 1102CML. W pierwszej kolejności zweryfikowano minimalny czas próbkowania i konwersji zastosowanego mikrokontrolera. Na wyjściu wybranego portu cyfrowego (RB5) mikrokontrolera na początku każdego cyklu timera ustawiano stan wysoki i rozpoczynano procedurę próbkowania i konwersji sygnałów z 8 (ze względu na n_{PD}) kanałów analogowych po czym załączano stan niski (Rys. 3.24a). Za pomocą oscyloskopu monitorowano poziom sygnału portu RB5. Długość zmierzonego impulsu jest całkowitym czasem operacji bloku ADC, który wyniósł $84\mu\text{s}$ na 8 kanałów czyli $10,5\mu\text{s}$ na 1 kanał. Krótszy czas uniemożliwiał konwersję w pełnym zakresie (napięcie po konwersji było niższe od napięcia na wejściu bloku ADC). Minimalizację czasu próbkowania i konwersji uzyskano przez ustawienie w rejestrze AD1CON3 wartości bitów ADCS=4 (zegar czasu konwersji) i SAMC=4 (czas automatycznego próbkowania).

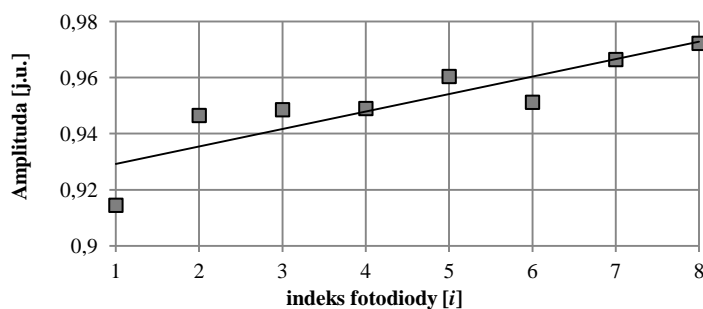
Następnie zweryfikowano czas ustalania sygnału na zastosowanej fotodiodzie TSL260RD. W eksperymencie zbliżono płaską przeszkodę do fotodiody tak, aby odbijała część światła generowanego przez diody LED lecz w tej odległości aby nie wzbudzać stanu nasycenia. Czas trwania pobudzenia (czas włączenia diody LED) ustawiono na 1ms. Zgodnie z dokumentacją elementu TSL260RD czas osiągnięcia 90% poziomu stanu ustalonego, mierząc od uzyskania poziomu 10% tego stanu wynosi ok. $260\mu\text{s}$ (Rys. 3.24b).



Rys. 3.24 Zrzuty ekranu z oscyloskopu Siglent. a) Długość impulsu potrzebnego na próbkowanie i konwersję 8 kanałów analogowych. b) Czas ustalania sygnału z fotodiody TSL260RD.

Czas ustalania sygnału jest ograniczeniem wskazującym na minimalny czas oczekiwania, t_{ocz} , przez który diody LED powinny być zapalone, aby sygnał na fotodiodach zdążył się ustalić. Ponieważ kanały analogowe próbkowane są przez mikrokontroler

sekwencyjnie, tylko przy pełnym ustaleniu stanu podczas konwersji pierwszej z fotodiod zarejestrowany zostanie właściwy profil intensywności światła. Kiedy t_{ocz} będzie zbyt krótki, pierwsza z próbkowanych fotodiod naładuje się do mniejszej wartości procentowej swojego zakresu w stosunku do ostatniej. W eksperymencie z ekranem odbijającym światło diod LED z powrotem do płaszczyzny fotodiod (ale tak aby nie osiągnąć nasycenia) sprawdzono stosunek poziomu sygnału dla poszczególnych fotodiod przy czasie $t_{ocz}=300\mu s$ do $t_{ocz}=500\mu s$ (Rys. 3.25). Eksperyment wykazał, iż dla $t_{ocz}=500\mu s$ kolejne fotodiody czujnika ładują się do bardziej zbliżonego poziomu niż dla $t_{ocz}=300\mu s$.



Rys. 3.25 Poziomy sygnałów sekwencyjnie próbkowanych wartości z 8 fotodiod po czasie $t_{ocz}=300\mu s$ w stosunku do $t_{ocz}=500\mu s$. Kierunek linii trendu uwiadczenia, iż dłuższy czas oczekiwania pozwala na naładowanie fotodiod do bardziej równomiernego poziomu.

Celem redukcji zapotrzebowania energetycznego sensora zdecydowano o ustaleniu czasu t_{ocz} na wielkość rzędu $300\mu s$. Wobec tego ustawienia używając rzeczywistego sensora drugiej realizacji do spróbkowania 8 kanałów analogowych, uwzględniając funkcję odpowiedzi skokowej fotodiody, ustalono czas włączenia diod LED na $t_{on}=375\mu s$, którą to wartość umieszcza się w liczniku ułamka przy wyznaczaniu parametru D (w mianowniku podaje się okres próbkowania, T_s). Dla konfiguracji sensora jak w jego fizycznej realizacji, biorąc pod uwagę ustalony czas t_{on} , oraz ilość fotodiod (8) i ich średni pobór prądu (1,1 mA), ilość diod LED (4) i ich średni pobór prądu (30,83 mA po zastosowaniu rezystorów $120\ \Omega$ i zasilaniu 5 V) obliczono pobór prądu przez elementy optoelektroniczne czujnika dla poszczególnych częstotliwości próbkowania. Wyróżniono sumaryczny pobór prądu przez diody LED, I_L , przez wszystkie elementy optoelektroniczne zakładając kluczkowanie również zasilaniem fotodiod (nie zrealizowano w fizycznej implementacji sensora), I_{TK} , oraz przez wszystkie elementy optoelektroniczne zgodnie z aktualną budową czujnika (kluczkowanie tylko zasilaniem diod LED), I_T . Podsumowanie prezentuje Tabela 3.5. Zużycie prądu przez mikrokontroler w czasie

przetwarzania zmierzono na 5,17 mA (25,85 mW). Optymalizacja tego poziomu nie była jednak jednym z głównych celów niniejszej rozprawy.

Tabela 3.5 Zużycie prądu przez czujnik o budowie liniowej dla różnych częstotliwości próbkowania układu.

f_s [Hz]	T_s [ms]	D [%]	I_L [mA]	I_{TK} [mA]	I_T [mA]
30	33	1,13	1,40	1,50	10,20
40	25	1,50	1,85	1,98	10,65
50	20	1,88	2,31	2,48	11,11
100	10	3,75	4,63	4,96	13,43
180	5,5	7,14	8,41	9,01	17,21
2666,67	0,4	100	115,6	123,9	124,43

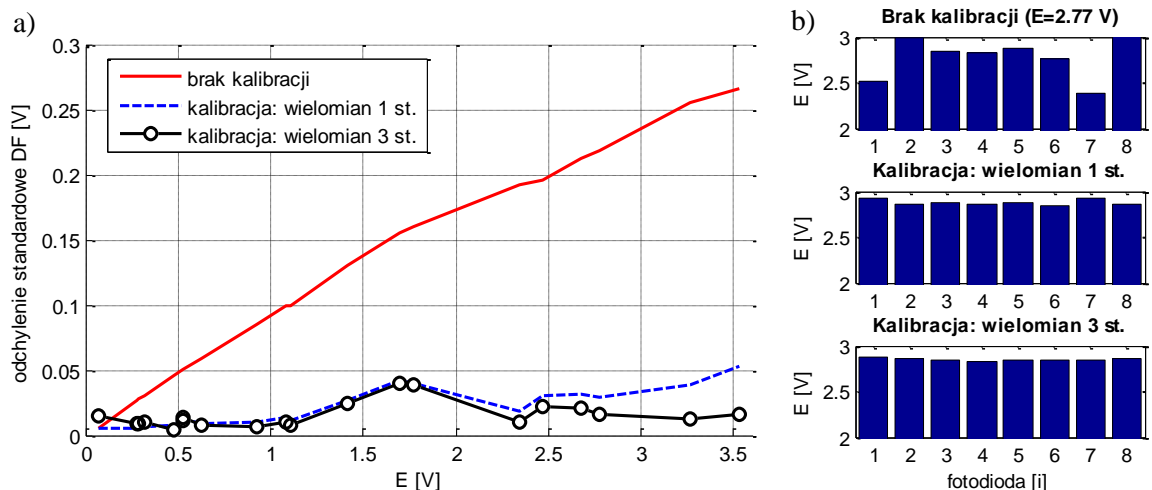
Średni poziom zużycia prądu przez pojedynczą fotodiodę TSL260RD (1,1 mA) oszacowano na podstawie pomiarów wykonanych przy różnym poziomie oświetlenia 8 fotodiod czujnika (Tabela 3.6). Jako urządzenie do referencyjnego pomiaru poziomu natężenia światła wykorzystano światłomierz Lutron LX-105.

Tabela 3.6 Pobór prądu przez fotodiody TSL260RD w zależności od poziomu natężenia światła padającego.

Warunki świetlne	Jasność [luksy]	Średni prąd poj. PD ($D=100\%$) [mA]	Prąd ośmiu PD ($D=1,5\%$) [μ A]	Moc ośmiu PD ($D=1,5\%$) [μ W]
Ciemność	0	0,959	115,05	575,25
Silne światło	872	1,125	135,00	675,00
Silniejsze światło	2000	1,235	148,20	741,00
Bezpośrednie nasłonecznienie	33500	1,041	124,95	624,75
średnia	-	1,090±0,12	130,80±14	654,00±71

Wykorzystanie mocy całego czujnika można zmniejszyć przez zastosowanie fotodiod innego typu (o krótszym czasie ustalania sygnału prowadzącym do niższego D), a także poprzez zaprojektowanie czujnika dla niższych napięć roboczych. Poziom zużycia prądu przez czujnik można również obniżyć m.in. przez zastosowanie adaptacyjnej częstotliwości próbkowania [153]. Kiedy czujnik przez dłuższy czas nie zaobserwował gestu, f_s może zostać zmniejszone co efektywnie zmniejszy wartość D i całkowity poziom zapotrzebowania na energię przez czujnik. Należy jednak wziąć pod uwagę, że gest wybudzający czujnik ze stanu o niższym poborze energii (czyli przywracający operacyjną częstotliwość próbkowania) może nie być prawidłowo rozpoznany. Może to wynikać z faktu, że prawdopodobnie nie będzie on zarejestrowany w całości.

Wykazane nieliniowości wynikające z czasów ustalania i konwersji sekwencji kanałów analogowych skorygowano razem z kalibracją czułości poszczególnych fotodiod czujnika. Czujnik umieszczono w zamkniętym kartonie i naprzeciwko niego w odległości ok. 10 cm umieszczono linijkę diod LED rozstawionych co 0,6 cm o łącznej szerokości 30 cm. Linijkę zasilono dwudziestoma różnymi poziomami napięcia, każde generujące inny poziom średniego natężenia światła odbierany przez sensor liniowy (w zakresie 0,1 - 3,6 V). Następnie, dla każdego z poziomów natężenia światła ustalono wartość średnią ze wszystkich fotodiod, która była oczekiwaną wartością do wskazania przez każdy z elementów. Na podstawie par wartości oczekiwanej i rzeczywistej, dla każdej z fotodiod wyznaczono funkcje kalibracyjne: wielomian pierwszego stopnia (funkcja liniowa) oraz wielomian trzeciego stopnia. Dla każdego średniego poziomu natężenia światła obliczono odchylenie standardowe wartości wskazywanych przez fotodiody w ramce danych (DF), zarówno bez kalibracji jak i dla dwóch uzyskanych funkcji kalibracyjnych (Rys. 3.26a). Dla wybranego poziomu natężenia światła (2,77 V) wykreślono także ramki danych przy braku kalibracji, kalibracji liniowej i za pomocą kalibracji wielomianem trzeciego stopnia odznaczające się odchyleniem standardowym DF kolejno o wartościach: 0,219 V, 0,029 V i 0,016 V (Rys. 3.26b)



Rys. 3.26 a) Odchylenie standardowe ramek danych bez kalibracji oraz po kalibracji wielomianami pierwszego i trzeciego stopnia. b) Ramki danych dla wybranego średniego poziomu natężenia światła odbieranego przez fotodiody ($E=2,77$ V).

Z Rys. 3.26a wynika, że wzrost poziomu natężenia światła odbieranego przez fotodiody powoduje wzrost zróżnicowania poziomu napięcia wyjściowego fotodiod czujnika (przebieg linii ciągłej). Zastosowanie kalibracji pozwala jednak na utrzymanie niskiego poziomu zróżnicowania wartości fotodiod praktycznie w całym zakresie operacyjnym

elementów (0 - 3,6 V). Ponadto, kalibracja za pomocą wielomianu trzeciego stopnia wykazała lepsze właściwości niż liniowa i to ona była stosowana w dalszych badaniach.

Szacuje się, iż zastosowanie funkcji kalibracji pozwoliło na korekcję trzech czynników: czułości danego egzemplarza fotodiody, różnic wynikających z rozpoczęcia momentu próbkowania w różnych punktach przebiegu ustalania sygnału poszczególnych fotodiod oraz wpływu potencjalnych niejednorodności w wielkości pola widzenia wynikających z możliwej niedokładności w umieszczaniu i przygotowywaniu nakładki bloków optycznych.

3.6. Możliwości zastosowania czujnika do obsługi gestów ciągłych

Gesty ciągłe polegają na reakcji systemu podejmowanej na bieżąco, w trakcie wykonywania gestu. Klasycznym tego przykładem jest odwzorowanie przemieszczenia fizycznej myszy komputerowej do ruchów kursora na monitorze. Rozważone zostaną dwa czynniki, które należy wziąć pod uwagę przy potencjalnym wykorzystaniu czujnika optycznego o liniowej budowie do intuicyjnej i komfortowej obsługi gestów ciągłych.

3.6.1. Nieliniowości odwzorowania położenia dłoni

Konstrukcja sensora o elementach rozłożonych umożliwia nie tylko hipotetyczne zarejestrowanie zdarzenia przesunięcia dłoni w polu widzenia sensora, rozróżnienie kilku pól dłoni lecz również, co wykazała analiza jego czułości przestrzennej, dokładne śledzenie lokalizacji dłoni w trakcie wykonywania gestu wzdłuż czujnika. Celem możliwie ciągłego śledzenia przemieszczenia dłoni wzdłuż osi x sensora należy w każdym cyklu próbkowania obliczyć wartość parametru COG . Jak pokazały rezultaty z punktu 3.5.5, przebieg funkcji parametru $shift$ zmierzonego (wzór 3.4) od rzeczywistego położenia obiektu jest funkcją monotoniczną dla obiektów poruszających się w *strefie detekcji* sensora, niezależnie od rodzaju pozy wykonującej przesunięcie (pod warunkiem, że jest to poza zwarta). Jednak w zależności od szerokości powierzchni odbijającej światło (ilość palców w zwartej pozycji) zaobserwować można różne stopnie nieliniowości odwzorowania składowej wzdłużnej pozycji obiektu. Na nieliniowość tę (widoczną przy krawędziach pola widzenia sensora) znaczny wpływ ma niejednorodne podświetlenie *strefy detekcji* przez diody LED (por. 0). Ma ono wpływ również na nieliniowości powstałe w osi y dla parametru max (maksymalna wartość profilu oświetlenia), jednak w tej osi dominującym czynnikiem jest

cecha spadku mocy pochodzącej od źródła optycznego uogólniając zgodna z prawem odwrotnych potęg (Rys. 3.15).

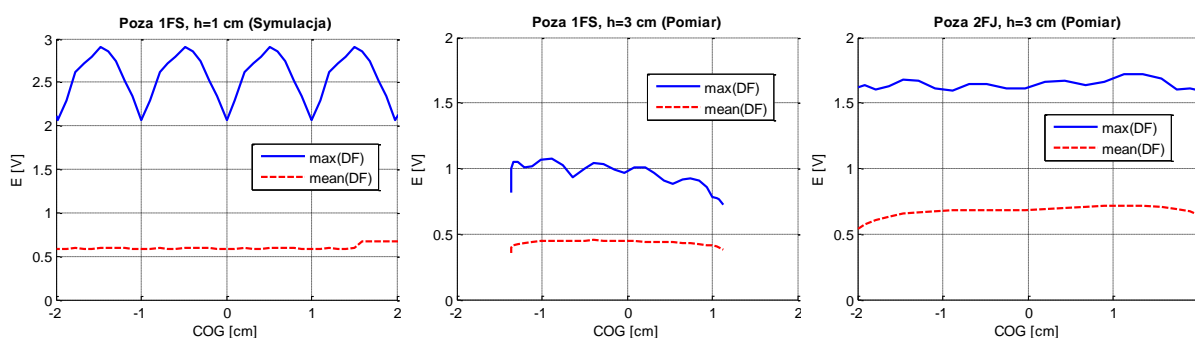
Potencjalna zdolność do śledzenia pozycji dłoni względem czujnika w sposób ciągły umożliwiająca potraktowanie czujnika jako interfejsu podobnego pod względem dwuosiowej lokalizacji do myszy komputerowej wymaga spełnienia dwóch warunków. Po pierwsze, chcąc zaadaptować wybrane parametry ramki danych (COG , max) jako współrzędne kursora na ekranie należy zredukować związane z nimi opisane nieliniowości funkcji parametru względem rzeczywistego położenia dłoni. Po drugie, częstotliwość próbkowania czujnika powinna być odpowiednio wysoka aby ruch na monitorze (np. odświeżanie lokalizacji kursora) był postrzegany jako płynny (np. 25Hz).

Na podstawie wyników symulacji i eksperymentów zdecydowano o zaleceniu używania pozy dłoni o dwóch palcach zwartych, 2FJ, do ruchów ciągłych ze względu na większą niż dla jednego palca (1FS) powierzchnię odbijającą światło (większy zasięg) i znacznie mniejsze nieliniowości niż w przypadku pozy pełnej dłoni (4FJ). W tym wypadku, najprostszym rozwiązaniem dla poprawy liniowości parametru COG jest ograniczenie wzdłużnego zakresu użytkowego, np. do $\pm 2,5$ cm (w tym zakresie, niezależnie od h w zakresie użytkowym standardowy błąd pozycji wzdłuż osi x praktycznie nie przekracza 0,1 cm, Rys. 3.23b). Dla częściowej redukcji nieliniowości w osi y możliwe jest zastosowanie korekcji przez funkcję postaci odwrotnej potęgi. Inną możliwością jest wykorzystanie sztucznych sieci neuronowych do zagadnienia regresji dwuwymiarowej i równoczesnej korekcji wartości obydwu parametrów (COG i max).

3.6.2. Utrzymanie dynamiki zmian amplitudy sygnału

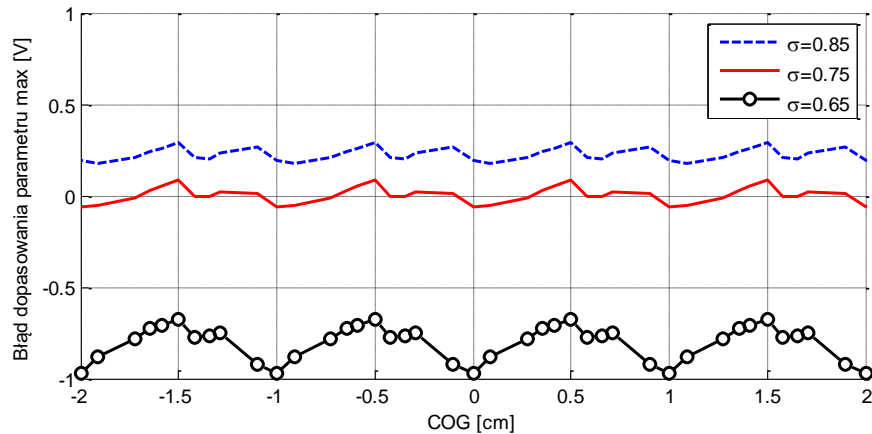
Obiekt znajdujący się w polu widzenia czujnika optycznego o budowie liniowej pracującego w trybie aktywnym przyczynia się do powstania na jego powierzchni profilu intensywności światła odbitego. Profil ten jest tym dokładniej pomierzony pod względem kształtu im większe zagęszczenie fotodiod na powierzchni czujnika. Idea używanego w niniejszej rozprawie czujnika polega jednak na rozproszeniu jego elementów światłoczułych (mała ilość fotodiod), toteż rejestrowany profil jest zgrubny. Skutkuje to faktem, iż w trakcie przemieszczania dłoni wzdłuż czujnika, w części okresów próbkowania może nie być rejestrowane rzeczywiste maksimum natężenia światła na jego powierzchni (bowiem w danym miejscu nie ma akurat fotodiody). Chcąc wykorzystać amplitudę sygnału jako wartość sterującą np. położeniem składowej pionowej kursora

na ekranie i zachowując dużą dynamikę sygnału należy wziąć pod uwagę zmienność parametru *max* ramki danych. Z wymienionych względów oraz z powodu nieidealnego profilu oświetlenia (niejednolicie promieniująca powierzchnia czujnika o rozłożonych diodach LED) będzie on o niestąlej (sinusoidalnej w pewnym zakresie przemieszczenia dłoni) wartości nawet podczas idealnie równoległego, w stosunku do czujnika, przesunięcia dłonią. Możliwością częściowego zniwelowania efektu sinusoidy jest obliczenie średniej wartości ze wszystkich fotodiod sensora, jednak kosztem zmniejszenia dynamiki śledzonych zmian położenia dłoni. W celu obserwacji opisanego efektu przeprowadzono badania symulacyjne i pomiarowe. Rys. 3.27a przedstawia parametry *max(DF)* oraz *mean(DF)* uzyskane na podstawie symulacji równoległego przemieszczenia poży 1FS do czujnika o 8 fotodiodach, na wysokości $h=1$ cm. Rys. 3.27b i Rys. 3.27c prezentują wyniki przeprowadzonych pomiarów dla poży 1FS i 2FJ przesuujących się wzdłuż czujnika na wysokości h ok. 3 cm.



Rys. 3.27 a) Wyniki symulacji dla poży 1FS przemieszczającej się przed czujnikiem na wysokości $h=1$ cm. b), c) Wyniki pomiaru dla poży 1FS oraz 2FJ przemieszczającej się przed czujnikiem na wysokości h ok. 3 cm. Przebiegi ciągle to wartość maksymalna ramki danych zaś przerywane to średnia z ramki danych.

Zaprezentowane wyniki wskazują na wpływ zarówno odległości poży od czujnika (Rys. 3.27a i Rys. 3.27b) jak i typu poży (Rys. 3.27b i Rys. 3.27c) na amplitudę sinusoidalnego kształtu parametru *max*. Z tego względu zaproponowana została metoda korekcji nieobserwowalnego maksimum amplitudy dla czujników o rozłożonych detektorach według procedury, której szczegółowy opis zawarty jest w jednej z publikacji autora niniejszej rozprawy [154]. Metoda bazuje na odpowiednim przekształceniu otrzymanego profilu intensywności światła odbitego i potraktowania go jako histogramu. Następnie wymaga ona obliczenia parametru odchylenia standardowego σ , które wyraża spiczastość rozkładu normalnego, najbardziej podobnego (pod względem minimum miary RSS) do danego histogramu. Metoda wykazuje dobre właściwości korekcji w badanym zakresie eksperymentów jest jednak bardzo czuła na niewielkie nieprawidłowości w określeniu parametru σ (Rys. 3.28).



Rys. 3.28 Przebiegi odtworzonej amplitudy sygnału dla różnych wartości σ (gdzie $\sigma=0.75$ oznacza idealne dopasowanie).

Zaproponowana korekcja amplitudy sygnału rejestrowanego w ramce danych wskazuje na możliwość poszerzenia zakresu użyteczności czujnika optycznego o budowie liniowej w kontekście nawigacji z wykorzystaniem gestów ciągłych.

3.7. Podsumowanie rozdziału

W pierwszej części rozdziału omówiona została zasada działania czujnika optycznego o liniowej budowie. W jej ramach zaprezentowana została idea dwóch trybów pracy, hipotetycznie przeznaczonych do rozpoznawania gestów w odmiennych warunkach natężenia światła otoczenia. Dla pomijalnego poziomu światła otoczenia dedykowany jest tryb aktywny, w którym fotodiody rejestrują światło odbite od dłoni a wygenerowane przez diody LED czujnika. Dla wysokiego poziomu natężenia światła otoczenia rozważany jest tryb pasywny, w którym fotodiody mierzą efektywnie stopień zacienienia powierzchni czujnika przez dłoń znajdującą się w jego pobliżu. W analizie literaturowej przeprowadzonej w rozdziale 2 nie znaleziono rozwiązań czujników optycznych charakteryzujących się możliwością zmiany trybu pracy w zależności od poziomu oświetlenia, zatem taka możliwość byłaby istotnym wkładem autora niniejszej rozprawy w rozwój metod interakcji za pomocą kilkudetektorowych czujników optycznych. Zadaniem badawczym jest zatem weryfikacja możliwości czujnika w rozpoznawaniu póż dłoni także w różnych warunkach oświetlenia, co zostanie zaprezentowane w rozdziale 4.

Przeprowadzenie pomiarów cech osobniczych na grupie 41 ochotników pozwoliło na weryfikację właściwości różnych konfiguracji sensora liniowego z uwzględnieniem wymagań podyktowanych cechami standardowego użytkownika o uśrednionych wymiarach dłoni i z uśrednionym poczuciem szybkości prędkiego i wolnego ruchu przesunięcia dłonią.

Do weryfikacji różnych konfiguracji czujnika optycznego a także zestawu zdefiniowanych w rozdziale parametrów wykorzystano zaprojektowany i wykonany do tych potrzeb symulator bazujący na metodzie Monte Carlo oraz zestaw pomiarowy zawierający fizyczną implementację czujnika.

Udowodniono, iż konfiguracja elementów optoelektronicznych sensora (1) ma zasadniczy wpływ na jego efektywny zakres *strefy detekcji*. Korzystna konfiguracja pozwala aby sygnał pochodzący z odbicia od obiektu zlokalizowanego blisko sensora ($h \sim 1$ cm) nie nasycał się i jednocześnie nie był za słaby dla obiektu umieszczonego bliżej górnej granicy *strefy detekcji* czujnika ($h \sim 5$ cm). Konfiguracja nr 1 elementów sensora posiada bardziej korzystne właściwości związane ze spłaszczeniem profilu podświetlenia niż rzeczywisty sensor, z udziałem którego przeprowadzono pomiary (ułożony według Konfiguracji nr 2). Możliwa jest zatem hipotetyczna poprawa zdolności czujnika do rozpoznawania gestów przy zastosowaniu rozłożenia elementów optoelektronicznych jak w Konfiguracji nr 1.

Rozważone zostały wymagania dotyczące wielkości poszczególnych stref (*martwa, niepewna, detekcji*) sensora o danej konfiguracji. Zapewniają one odpowiedni zakres *strefy detekcji* (od $h \geq 1$ cm) niezależnie od trybu pracy sensora (pasywny / aktywny) oraz potencjalną możliwość rejestracji pełnego profilu natężenia światła pochodzącego od całej dłoni (poza 4FJ), umieszczonej centralnie nad sensorem (dzięki odpowiednio szeroko rozłożonym fotodiodom). Każdy z czujników zbudowanych według zaproponowanych w ramach poszczególnych krotek formacji elementów jest w stanie wykrywać (obserwować obydwie krawędzie przekroju) objekty o szerokości do 8 cm (Rys. 3.20b), znajdujące się centralnie nad nimi a zatem większe niż wynosi szerokość standardowej dłoni według pomiarów przeprowadzonych na grupie ochotników ($2 \cdot D_2 = 6,4$ cm, Tabela 3.2).

Wykazano, że zastosowanie bloków optycznych znacznie poprawia zdolność rozdzielczą liniowego sensora optycznego. Rozpatrzono standardową pozę 2FS (o wymiarach ustalonych na podstawie pomiarów grupy ochotników) dla różnie skonfigurowanych czujników. Symulacje wykazały, iż palce ułożone w takiej pozie, przy założeniu kryterium minimalnej wartości $I_D = 0,15$ powinny być rozróżnialne niemal w całym zakresie *strefy detekcji* czujnika ($1 \text{ cm} \leq h \leq 5 \text{ cm}$) przez czujnik skonfigurowany jak w krotce 3. Rzeczywista implementacja czujnika pozwala na rozróżnienie palców w nieco mniejszym zakresie odległości ($1 \text{ cm} \leq h \leq 3,7 \text{ cm}$). Separacja w osi x jest możliwa

w całym zakresie widoczności palców, czyli dopóki jeden z palców nie wyjdzie poza *strefę detekcji* sensora. Zdolność rozdzielcza czujnika o budowie liniowej jest determinowana przez parametry geometryczne przesłon optycznych ustalających pole widzenia elementów optoelektronicznych.

Obecność bloków (przesłon) optycznych ograniczających pole widzenia poszczególnych fotodiod wpływa zatem na odpowiednią wartość zdolności rozdzielczej czujnika i tym samym na możliwość rozróżniania przez niego palców w pozie 2FS, odróżniając ją od póż o zwartych palcach, dla dłoni znajdującej się w *strefie detekcji* czujnika. Rozróżnianie dwóch póż (2FS i pozy zwartej, sekcja 3.5.4) pozwala na częściowe udowodnienie pierwszej z tez postawionych w niniejszej rozprawie doktorskiej:

Możliwe jest opracowanie kilkuelementowego czujnika optycznego zbudowanego z liniowo rozłożonych fotodiod oraz diod LED, jak również przesłon optycznych formujących pole widzenia fotodiod, umożliwiającego rozpoznawanie co najmniej 4 póż dłoni.

Wprowadzenie metody rozróżniania póż zwartych (1FS, 2FJ i 4FJ) pozwoli na uzyskanie czujnika klasyfikującego w sumie 4 różne pozy dłoni (wliczając pozę 2FS) i tym samym udowodnienie tezy pierwszej całkowicie. Dodatkową możliwością poszerzenia zbioru rozpoznawanych póż jest rozróżnianie ilości palców w pozie 3 i 4 palców rozpostartych, lecz ze względów na fizjonomiczną niewygodę oraz właściwości i rozmiary zastosowanego w badaniach czujnika nie prowadzono badań w zakresie ich wykrywania.

W rozdziale pokazano również, iż czujnik optyczny o liniowo rozłożonych detektorach jest zdolny do śledzenia pozycji zwartej pozy dłoni poruszającej się w jego polu widzenia dzięki wysokiej czułości przestrzennej. W zależności od pozy, różny jest stopień zniekształceń nieliniowych funkcji pozycji pomierzonej (parametr *COG*) względem rzeczywistej. Jednak zwłaszcza w ograniczonym zakresie ($shift = \pm 2,5$ cm) odchylenie standardowe położenia jest na tyle niewielkie (poniżej 0,1 V niezależnie od *h*), iż umożliwia rozpatrzenie czujnika jako interfejsu gestów ciągłych. Poza dłoni 2FJ została zalecona do wykonywania gestów ciągłych jako kompromis między stopniem nieliniowości funkcji położenia a wielkością powierzchni odbijającej (dzięki czemu jest lepiej widoczna dla czujnika niż poza 1FS i lepiej odwzorowywana jest jej lokalizacja niż w przypadku pozy 4FJ).

Czułość czasowa czujnika definiowana jest przez jego częstotliwość próbkowania. Wskazane zostały dwa parametry: C , współczynnik wykrywalności ilości przesunięć oraz D , współczynnik wypełnienia okresu próbkowania stanem włączonym diod LED, od którego zależy poziom poboru prądu. Wzrost wartości częstotliwości próbkowania powoduje wzrost parametru C i przy $f_s=100\text{Hz}$ wykrywanych jest ponad 98% przesunięć. Jednak dla częstotliwości próbkowania o połowę mniejszej, która obniża także dwukrotnie zużycie prądu przez elementy optoelektroniczne czujnika, wartość C spada stosunkowo niewiele, gdyż o 14p.p. W badaniach dotyczących rozpoznawania póż, które zostaną opisane w następnym rozdziale, częstotliwość próbkowania nie ma wpływu na dokładność klasyfikacji gdyż z definicji podlega jej tylko jedna obserwacja sygnału wejściowego. Natomiast w badaniach związanych z rozpoznawaniem gestów, gdzie analizuje się sekwencje i gdzie ważne są relacje pomiędzy poszczególnymi położeniami dłoni i ilość obserwacji w trakcie jednego gestu, zastosowano wyższą wyjściową częstotliwość próbkowania, $f_s=100\text{Hz}$.

Celem ustalenia minimalnego dopuszczalnego czasu włączenia diod LED (wpływającego na wartość współczynnika D) zmierzono czas potrzebny na ustalenie sygnału w zastosowanej fotodiodzie TSL260RD oraz na sekwencyjne próbkowanie wyjść ze wszystkich fotodiod przez wykorzystany mikrokontroler PIC24FV16KA302. Wynikowy dopuszczalny czas stanu włączenia diod LED, t_{on} , został ustalony na $375\mu\text{s}$. Opracowana została tabela zawierająca zużycie prądu przez elementy optoelektroniczne sensora liniowego, o konfiguracji i parametrach jak w fizycznej implementacji sensora, przy założeniu różnych częstotliwości próbkowania (Tabela 3.5). Wynikowe zużycie prądu i mocy przez elementy optoelektroniczne czujnika gestów o liniowej budowie, przy zastosowaniu optymalnej konstrukcji (kluczowanie zasilania fotodiod), zarówno dla 50Hz (2,48 mA; 12,4 mW) jaki i 100Hz (4,96 mA; 24,8 mW) jest porównywalne z zapotrzebowaniem raportowanym przez autorów innych podobnych rozwiązań czujników optycznych zaprezentowanych w literaturze (np. 3,78 mA [54], od 7,2 mW do 29 mW [65]). Zatem hipotetycznie większa liczba gestów rozpoznawanych przez projektowany sensor o budowie liniowej nie będzie ponoszona kosztem proporcjonalnie większego zużycia energii niż w innych czujnikach.

Ponadto zademonstrowano, iż dzięki liniowemu rozłożeniu elementów optycznych czujnik posiada potencjał nie tylko do rozróżniania póż dłoni lecz i do śledzenia zmieniającej

się lokalizacji zwartej pozy dla dłoni znajdującej się w polu widzenia sensora. Miary oceny położenia dłoni (środek ciężkości profilu oświetlenia, *COG*, oraz średnia bądź maksimum profilu oświetlenia) umożliwiają użytkownikowi nawigację w sposób ciągły w dwóch osiach (prostopadłej i równoległej do sensora) odzwierciedlając np. część funkcjonalności myszy komputerowej. Zaproponowany algorytm śledzenia rzeczywistej wartości maksimum profilu natężenia światła odbitego, niezależnie od pozycji dłoni względem sensora w obrębie *strefy detekcji*, pozwala na utrzymanie wysokiej dynamiki odwzorowywania ruchów dłoni w osi *y* na zmiany widoczne w interfejsie użytkownika (np. pozycja kursora w osi pionowej ekranu) lecz wymaga on jeszcze zastosowania korekcji nieliniowości.

Dzięki przeprowadzonym badaniom eksperymentalnym i symulacyjnym potwierdzona została zdolność czujnika optycznego do rozróżniania zwartych i rozpostartych póz dłoni, spełnione zostały założenia co do wielkości zakresu operacyjnego i *strefy detekcji* oraz osiągnięto poziom zużycia energii przez elementy optoelektroniczne na poziomie porównywalnym z innymi tego typu czujnikami zaprezentowanymi w literaturze. Wnioski te pozwalają rozważyć projektowany czujnik jako potencjalnie atrakcyjne urządzenie umożliwiające prywatną (dyskretną) interakcję człowieka z urządzeniami mobilnymi. W związku z tym, w następnym rozdziale sprawdzona została dokładność metod w rozpoznawaniu póz i gestów dłoni na podstawie sygnałów pochodzących z czujnika optycznego o liniowej budowie.

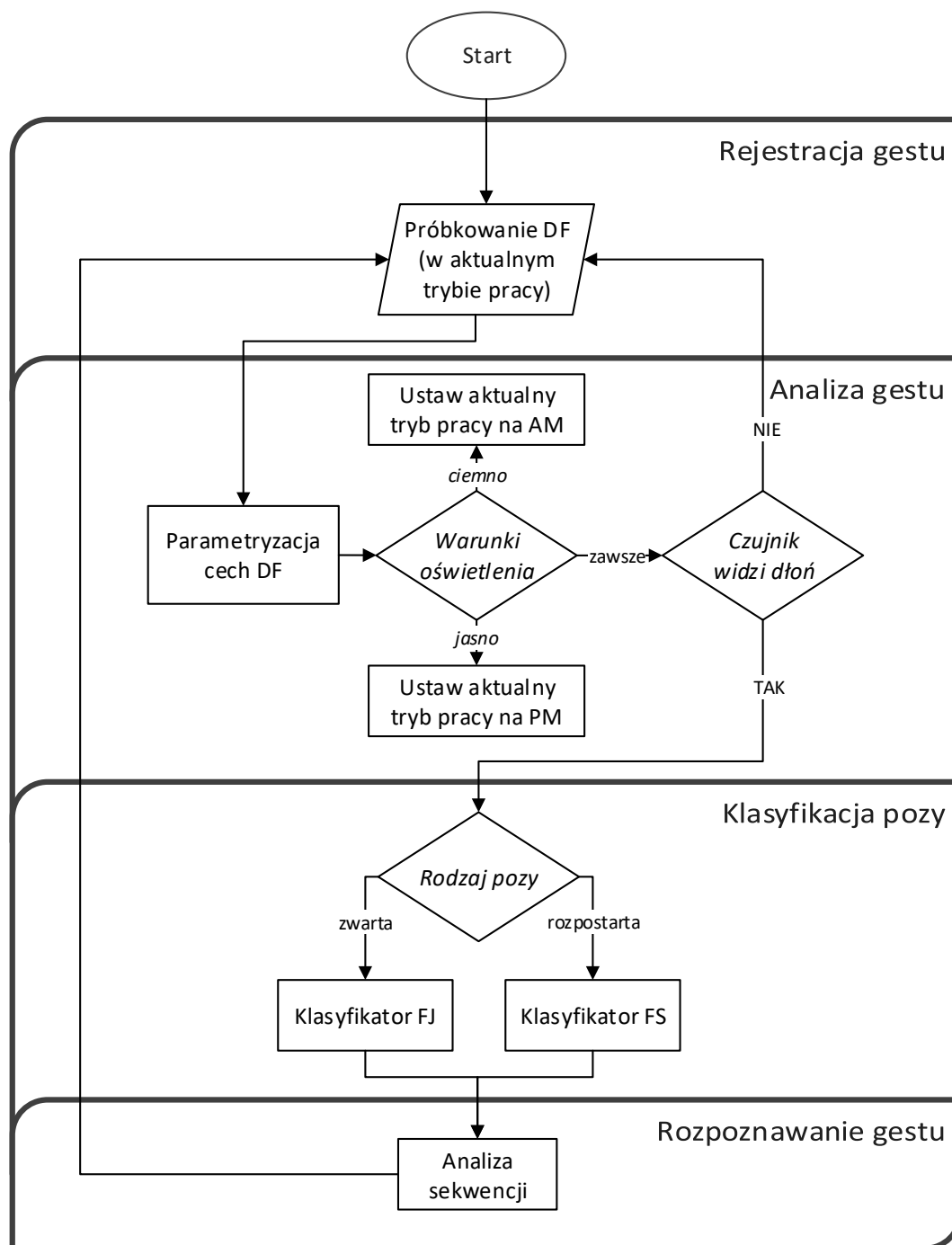
4. KLASYFIKACJA PÓZ I GESTÓW Z UŻYCIEM OPTYCZNEGO CZUJNIKA O LINIOWEJ BUDOWIE

W pierwszej części niniejszego rozdziału opisano zaadaptowaną strukturę systemu rozpoznawania gestów. W dalszej kolejności zweryfikowano, czy możliwe jest uzyskanie wysokiej dokładności klasyfikacji pojedynczych odczytów profili natężenia światła powstałych przy udziale różnych póz dłoni z pomocą wybranych metod. Następnie sprawdzono jak wybrana metoda klasyfikacji póz oraz pozostałe właściwości czujnika o budowie liniowej zależą od różnych warunków oświetlenia i trybów pracy sensora. Zaprojektowano i przetestowano również modele klasyfikacji sekwencji póz składających się na dany gest z uwzględnieniem stopnia przetworzenia ramki danych i różnych częstotliwości próbkowania sensora.

4.1. Implementacja GRS

Według przytoczonej w rozdziale 2 definicji systemu rozpoznawania gestów, GRS, na proces klasyfikacji gestu może składać się kilka faz. Mogą one zostać opisane jako funkcjonalne komponenty kontrolera GRS. Koncepcja ta, sformułowana przez Berman i Sterna [34], została zaadaptowana przez autora rozprawy do użycia w czujniku optycznym (Rys. 2.5). W ramach kolejnych faz przetwarzania sygnałów wejściowych przez GRS zaproponowano poszczególne bloki funkcjonalne i decyzyjne (Rys. 4.1).

Początkowa faza ma miejsce podczas rejestracji gestu, po której następują etap analizy, potencjalna klasyfikacja pozy dłoni oraz rozpoznanie całego gestu. Ostatni z etapów jest najczęściej realizacją Maszyny Stanów, Modeli Markowa, RNN lub innych metod [108][137][155]. Założeniem schematu jest jedno próbkowanie w każdym cyklu lecz możliwe jest także rozważenie trybu pracy przeprowadzającego próbkowanie w dwóch trybach w jednym cyklu (wtedy pomijany jest blok analizy warunków oświetlenia).



Rys. 4.1 Schemat zaadaptowanego systemu rozpoznawania gestów GRS z wyszczególnionymi kolejnymi fazami przetwarzania sygnałów wejściowych (rejestracja gestu, analiza gestu, klasyfikacja pozy, rozpoznawanie gestu).

Rejestracja całego gestu, czyli niewerbalnej informacji trwającej w czasie, wymaga przeprowadzenia sekwencji próbkowań obserwacji jego poszczególnych etapów. Podczas pojedynczego okresu próbkowania możliwa jest rejestracja profilu natężenia światła w trybie pasywnym, PM (ang. *Passive Mode*, bez włączonych diod LED), aktywnym, AM (ang. *Active Mode*, z włączonymi diodami LED), lub w trybie różnicowym, DM (ang. *Differential Mode*, sekwencyjne próbkowanie w trybach AM i PM i obliczenie różnicy między ramkami danych). Czas potrzebny na przeprowadzenie dodatkowego próbkowania w trybie DM wynosi ok. $100\mu\text{s}$ ($84\mu\text{s}$ na próbkowanie i konwersję oraz czas przełączania diod LED przez zmianę stanu sygnału na porcie cyfrowym), zatem ręka przemieszczająca się z prędkością 100 cm/s ulegnie przemieszczeniu jedynie o $0,01\text{ cm}$ co jest wystarczająco małym błędem aby rozważyć ten sposób rejestracji. Tego typu próbkowanie, przez odjęcie od siebie profili AM i PM, hipotetycznie pozwoli na wyeliminowanie składowej światła otoczenia. Badania nad określeniem kryterium bloku decyzyjnego *Warunki oświetlenia* determinującym preferowany tryb pracy czujnika zamieszczono w podrozdziale 4.3 rozprawy.

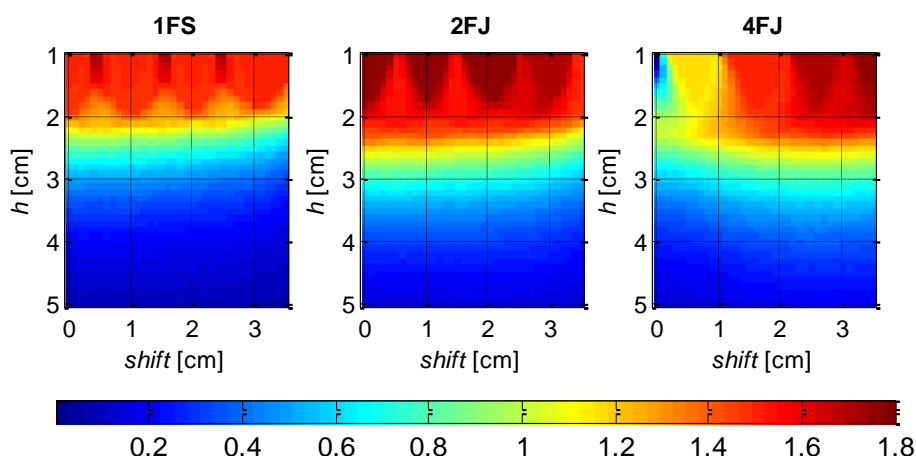
Dokładna funkcjonalność poszczególnych bloków zaproponowanego GRS zależy będzie od wymagań wybranych metod rozpoznawania póż i gestów oraz ustalonych kryteriów (bloków decyzyjnych). Mogą one różnić się typem danych wejściowych do modelu, ich rozmiarem czy stopniem przetworzenia (np. do wektora cech). Z tego względu w pierwszej kolejności rozważone zostaną metody klasyfikacji póż dłoni (faza *Klasyfikacja pozy*), które determinować będą wymagane działania przygotowujące dane (faza *Analiza gestu*).

Wyjątkiem jest rozważenie bloku odpowiadającego za decyzję czy w danym cyklu próbkowania czujnik zauważył profil na tyle nierównomierny, iż prawdopodobnie zarejestrował on więcej niż sygnał tła (blok: *Czujnik widzi dłoń*). *Strefa detekcji* czujników typu zbliżeniowego, w której rejestrują one przemieszczające się w ich polu widzenia obiekty, często ograniczona jest do kilkunastu czy kilkudziesięciu centymetrów [39]. Jednak w kontekście nawigacji mobilnych urządzeń osobistych, a zwłaszcza okularów multimedialnych (lub, w mniejszym stopniu gogli VR) gesty wykonywane są często w polu bliskim czujnika (do 10 cm). W przypadku czujnika optycznego o liniowej budowie wpływ zmian w rejestrowanym profilu natężenia światła generowany przez obiekty przemieszczające się poza jego założoną *strefą detekcji* ($1\text{ cm} \leq h \leq 5\text{ cm}$)

może być marginalny ze względu na charakterystykę czujnika, na którą składa się jego rozdzielczość kątowna (β_{PD}), czułość fotodiod i moc diod LED. Z przetworników stosowanych w detektorach gestów jako nieprzetworzone dane pomiarowe (ang. *raw data*) otrzymuje się przebieg z jednego lub kilku rozłożonych elementów rejestrujących. Pojawienie się obiektu w *strefie detekcji* czujnika, generującego zmiany sygnału mierzonego przez czujnik, powoduje zmiany w statystycznym opisie rejestrowanego przebiegu. W przypadku czujnika z jednym detektorem może to być zmiana sygnału w czasie, zaś w przypadku czujnika rozłożonego (kilkudetektorowego) oprócz zmian w czasie, obserwować można także, w danym momencie, przestrzenną równomierność profilu natężenia rejestrowanego światła. Często stosowanym parametrem statystycznym wykorzystywanym w tym celu jest wariancja, σ^2 (lub odchylenie standardowe, σ). Wartość wariancji (lub odchylenia standardowego) danego parametru w czasie (lub ramki danych przestrzennych) w relacji do konkretnego progu jest stosowanym w literaturze kryterium na rozpoczęcie i zakończenie fazy rejestracji danych do analizy przez GRS [32][48][119].

Dla rozważanego w niniejszej rozprawie czujnika gestów o liniowej budowie jako kryterium równomierności profilu rejestrowanego natężenia światła obliczane jest parametr odchylenia standardowego ramki danych, $sd(DF)$. Zakładając tryb aktywny pracy czujnika, wartość $sd(DF)$ bliska 0 V oznacza najpewniej brak obiektu odbijającego światło w *strefie detekcji* sensora i brak konieczności analizy gestu. Symulacje trybu aktywnego czujnika z wykorzystaniem symulatora opracowanego w rozdziale 3.4.2 pozwoliły na uzyskanie poglądowego rozkładu wartości $sd(DF)$ w zależności od położenia (x_o, y_o) danej pozy (1FS, 2FJ lub 4FJ) względem czujnika (Rys. 4.2). Rozważano położenia dłoni zlokalizowane w *strefie detekcji* czujnika ($1\text{ cm} \leq h \leq 5\text{ cm}$) i znajdujące się co najwyżej nad jego najbardziej skrajną fotodiodą ($shift \leq 3,5\text{ cm}$). Wielkości opisujące poszczególne pozy są równe tym, które zastosowano do wcześniejszych symulacji. Z wykresów przedstawionych na Rys. 4.2 wynika, iż niemal dla każdej z rozważanych lokalizacji póź uzyskana wartość $sd(DF)$ przekraczała poziom 0,1 V. Jedynie dla pozy 1FS oddalonej od czujnika ($h > \sim 4\text{ cm}$) oraz pozy 4FJ położonej centralnie nad czujnikiem i bardzo blisko niego ($h \sim 1\text{ cm}$), gdzie uzyskiwany profil światła odbitego był niemal jednorodny lecz o dużej wartości średniej, uzyskano wartość $sd(DF)$ poniżej 0,1 V. Dla sensora pracującego w trybie aktywnym, na podstawie wykonanych symulacji i eksperymentów zdecydowano zatem o ustaleniu

wartości progowej odchylenia standardowego ramki danych na $sd_{Ta}=0,1$ V [156][157], jedynie po przekroczeniu której będzie ona analizowana pod kątem klasyfikacji pozy.



Rys. 4.2 Symulacje wartości $sd(DF)$ w funkcji pozycji pozy względem czujnika optycznego o konfiguracji jak w rzeczywistej implementacji czujnika dla różnych póz zwartych.

Określenie progu sd_T pozwala na zautomatyzowaną i efektywną rejestrację profili natężenia światła powstałych w wyniku odpowiednio blisko zlokalizowanej dłoni przed sensorem przez minimalizację zapisu niemiernodajnych przykładów (zbyt płaski profil, niskie $sd(DF)$) co jest istotne dla metod uczenia maszynowego. Badania związane z określaniem wartości progu dla trybu pasywnego (sd_{Tp}) zostały opisane w rozdziale 4.3.

4.2. Klasyfikacji póz

W niniejszej rozprawie rozważone zostały dwie metody klasyfikacji póz dłoni z wykorzystaniem danych rejestrowanych przez czujnik optyczny o liniowej budowie. Domyślnym trybem pracy czujnika, w kontekście badania metod rozpoznawania póz dłoni, jest tryb aktywny. Część pracy związana z detekcją póz w trybie pasywnym zostanie odpowiednio podkreślona.

4.2.1. Baza póz dłoni

Projekt i ewaluacja metod klasyfikacji póz dłoni wykonane mogą być na odpowiednio przygotowanym zbiorze danych. W tym celu poproszono 11 ochotników (7 mężczyzn, 4 kobiety; wiek od 23 do 45 lat; R od 7 do 10 mm) o wzięcie udziału w eksperymencie. Czujnik gestów pracujący w trybie aktywnym, z $f_s=10$ Hz, umieszczony został w specjalnym uchwycie, stabilnie względem stołu pomiarowego, fotodiodami do góry. Konstrukcja została umieszczona w kartonie o 5 ścianach aby zredukować potencjalny wpływ zmiany poziomu natężenia światła otoczenia. Jedyne otwory w kartonie znajdowały się od boku równoległego

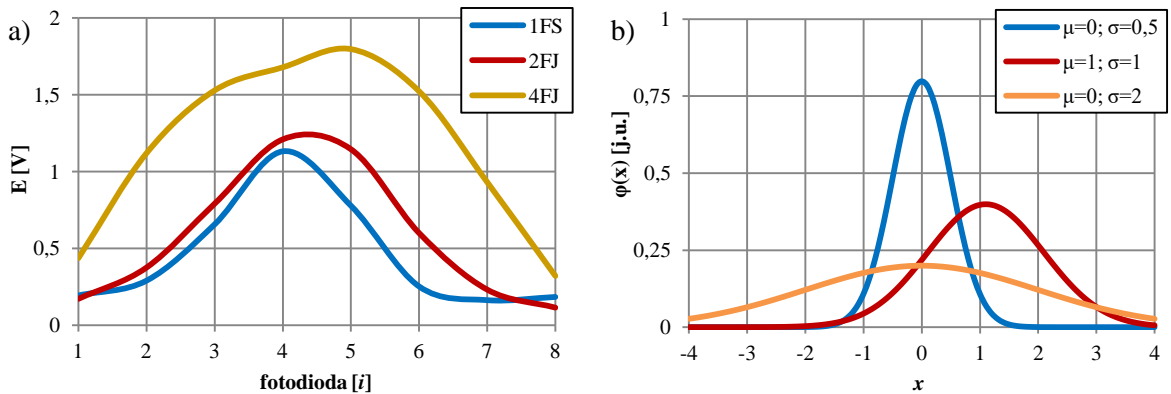
do osi x sensora. Ochotnicy zostali poproszeni o uformowanie dłoni kolejno w pozy 1FS, 2FJ, 4FJ i powtórzenie określonej procedury dla każdej z nich. Każda osoba miała za zadanie umieścić dłoń z daną pozą w polu widzenia czujnika liniowego i przemieszczać ją w losowe lokalizacje w obrębie *strefy detekcji* czujnika, aż zarejestrowanych zostanie 200 ramek danych, po czym zabrać dłoń z pola widzenia czujnika. Czujnik na bieżąco wysyłał dane przez interfejs UART do komputera gdzie specjalnie napisana w tym celu aplikacja tworzyła dla każdego użytkownika osobny plik zaś zestaw ramek danych z różnych gestów zapisywany był w osobnych arkuszach tego pliku. Przelączenie zapisu do nowego arkusza dedykowanego kolejnej pozie odbywało się ręcznie, w przerwach pomiędzy zmianą pozy dłoni. Aplikacja kontrolowała, aby dla każdej pozy zapisanych zostało 200 ramek danych spełniających warunek $sd(DF) > sd_{Ta}$. Podczas projektowania procedur potrzebnych do utworzenia tego zbioru danych rozważano jedynie pozy dłoni o zwartych palcach, stąd brak w nim póź o kodzie 2FS (dwa palce rozpostarte), których obiektywna dokładność rozpoznawania będzie określona przez model klasyfikacji gestów prezentowany w dalszej części rozprawy.

Wynikowo zarejestrowano 6600 przykładów, po 2200 na każdą z póź. Dzięki kontrolowanemu zapisowi póź, możliwe jest stowarzyszenie każdej ramki danych z etykietą (nazwą klasy) oznaczającą pozę która ją tworzy. Informacja ta pozwala na przeprowadzenie metod klasyfikacji z grupy uczenia nadzorowanego. Dodatkowo, już w trakcie analiz zdecydowano o ograniczeniu wymagań rozpoznawania kształtu przez metody klasyfikacji. Obostrzenie polegało na tym aby analizować jedynie profile, w których wartość maksymalna nie jest obserwowana przez żadną ze skrajnych fotodiod ($1 < IM < n_{PD}$, gdzie IM to indeks fotodiody o największej wartości sygnału w danej DF). Dłoń zlokalizowana nad krawędzią sensora może bowiem być źródłem niejednoznacznych profili (trudno odróżnić dwa skrajne palce pozy 4FJ od całej pozy 2FJ dla tak umieszczonej względem czujnika dłoni). W efekcie tego obostrzenia rozkład licznosci klas przestał być jednak równomierny gdyż otrzymano następującą licznosc obserwacji kolejnych klas (1FS, 2FJ, 4FJ): 1859, 2040, 2154. Uzyskany zbiór danych podzielono na trzy rozłączne zbiory: treningowy, walidacyjny i testowy o proporcjach 70%/15%/15%.

Warunki na wartość IM oraz na wartość odchylenia standardowego ramki danych mającej przekroczyć ustalony dla danego trybu pracy próg (sd_{Ta} lub sd_{Tp}) ustanowiono jako kryteria bloku decyzyjnego GRS o nazwie *Czujnik widzi dłoń w fazie Analiza gestu*.

4.2.2. Metoda I: dopasowanie krzywych wzorca i obserwacji

W trakcie wczesnych etapów ewaluacji i eksperymentów związanych z badaniem odpowiedzi optycznego czujnika gestów na różnego rodzaju pobudzenia (różne zwarte pozy dłoni umieszczone w różnych miejscach pola widzenia czujnika) zwrócono uwagę na wizualne podobieństwo kształtów rejestrowanych profili natężenia światła, k_O (krzywa obserwacji) (Rys. 4.3a) do kształtów ogólnych rozkładów normalnych, k_W (krzywa wzorca), o różnej wartości odchylenia standardowego σ (Rys. 4.3b).



Rys. 4.3 a) Profile intensywności światła odbitego, k_O , powstałe w wyniku ułożenia palców dłoni w różne pozy w polu widzenia sensora. Przebiegi uzyskano drogą pomiaru. b) Funkcje rozkładu normalnego, k_W , o różnych wartościach odchylenia standardowego (σ) i średniej (μ).

Celem rozpatrzenia zagadnienia hipotetycznego podobieństwa krzywych w rozprawie wykorzystano następujący wzór na wartość gęstości prawdopodobieństwa rozkładu normalnego w punkcie x_i :

$$k_W(x_i|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right). \quad (4.1)$$

Standardowy rozkład normalny (w odróżnieniu od ogólnego) opisują konkretne wartości: $\mu=0$ oraz $\sigma=1$. W dalszej części rozprawy wykorzystywany będzie ogólny rozkład normalny. Analizując wielkość parametru σ , odchylenia standardowego rozkładu, widoczna jest jej proporcjonalność do smukłości rozkładu (Rys. 4.3b). Dzięki tej znanej własności, sterując wartością tego parametru, hipotetycznie możliwe jest uzyskanie rozkładu normalnego, który będzie bardziej podobny smukłością do jednego z profili natężenia światła pochodzącego z rozważanych póz dłoni niż do pozostałych. Wartość średnia rozkładu, μ , w graficznej reprezentacji przesunęła cały rozkład w prawo ($\mu>0$) lub lewo ($\mu<0$) względem początku układu współrzędnych. Wielkość x_i we wzorze oznacza współrzędną i -tej próbki, dla której obliczana jest wartość rozkładu.

Zdecydowano zatem o próbie opisu profili pochodzących od trzech typów pól zwartych (1FS, 2FJ, 4FJ) przez funkcje rozkładu normalnego. Pożądanym efektem będzie pogrupowanie opisów rozkładów do trzech, możliwie wykluczających się klas.

Jedną z własności rozkładu gęstości prawdopodobieństwa jest, iż pole pod jego wykresem równa się jedności. Własność ta musi być zachowana bez względu na ilość próbek (w przypadku reprezentacji dyskretnej) przybliżających rozkład. Z tego powodu, celem opisanego krzywej k_O o n_{PD} próbkach (o wartościach wyrażonych w woltach) za pomocą parametrów rozkładu normalnego, należy dany profil znormalizować tak, iż suma jego wartości także będzie równa jedności:

$$k_{Oi} = \frac{v_i}{\sum_{i=1}^{n_{PD}} v_i}, \quad (4.2)$$

gdzie k_{Oi} to i -ta próbka krzywej obserwacji, v_i to poziom natężenia światła na i -tej fotodiodzie wyrażone w woltach. Następnie, biorąc pod uwagę liczbę fotodiod sensora rejestrującego profil, n_{PD} , oraz ich rozstawienie, d_{PD} , należy obliczyć pozycję każdej i -tej fotodiody względem początku układu współrzędnych, które to będą wykorzystane jako współrzędne próbek w rozkładzie wzorca:

$$x_i = -\frac{(n_{PD}-1)}{2}d_{PD} + (i-1)d_{PD}. \quad (4.3)$$

Zdefiniowany wcześniej w rozprawie parametr środka ciężkości profilu intensywności światła, COG , można potraktować jako wartość μ rozkładu wzorca. Dzięki temu, środek symetrii rozkładu wzorca będzie podążać za środkiem ciężkości obserwowanego profilu światła, którego zmiany położenia powodowane są przemieszczaniem dłoni wzdłuż osi x . Znając także położenia poszczególnych punktów próbkowania, x_i , zagadnienie opisu profili natężenia światła funkcjami rozkładu normalnego sprowadza się do znalezienia trzech wartości σ , dla których krzywe k_O i k_W związane z daną pozą będą do siebie najbardziej podobne. Zdecydowano, iż stopień podobieństwa będzie obliczany stosując miarę rozbieżności RSS:

$$RSS(k_O, k_W) = \sum_{i=1}^{Ns} (k_{Oi} - k_{Wi})^2, \quad (4.4)$$

gdzie Ns to ilość obserwacji krzywej (próbek), k_{Oi} oraz k_{Wi} to kolejno i -ta próbka krzywej obserwacji i wzorca. W jednej z publikacji autora rozprawy ([47]) miara odległości była jeszcze dzielona przez wariancję k_O , lecz ze względu na to, iż wartość ta jest taka sama

dla każdej z trzech par i jedynie skaluje wyniki, została ona pominięta. W praktyce wartość N_s jest równa ilości fotodiod czujnika, n_{PD} , toteż modelowany rozkład normalny powinien być obliczany w n_{PD} punktach. W trakcie klasyfikacji zatem, do każdej otrzymanej krzywej obserwacji k_O , dopasowane zostaną trzy krzywe wzorca k_W , każda związana z opisem innej pozy (przez zróżnicowane wartości σ rozkładu normalnego). Poza, która utworzyła krzywą obserwacji k_O zostanie rozpoznana dzięki największemu podobieństwu z daną krzywą k_W . Największe podobieństwo wykazywane jest dla pary krzywych obserwacji i wzorca generujących najmniejszą wartość miary rozbieżności RSS (wzór 4.4):

$$\text{kod rozpoznanej pozy} = \arg \min_l (RSS(k_O, k_W(x_i|COG, \sigma_l))) \quad (4.5)$$

gdzie dla $l=1$: $\sigma_l = \sigma_{IFS}$, dla $l=2$: $\sigma_l = \sigma_{2FJ}$, dla $l=3$: $\sigma_l = \sigma_{4FJ}$. W ramach określenia właściwych wartości odchylenia standardowego opisujących każdą z trzech klas zwartych pód dłoni (σ_{IFS} , σ_{2FJ} , σ_{4FJ}) wykonano trzy podejścia. Stosowane w poszczególnych podejściach symulacje i różne realizacje czujników optycznych wynikały z równoczesnego, w stosunku do prowadzonych badań, rozwoju koncepcyjnego i konstrukcyjnego czujnika gestów oraz rozwoju modelu symulatora zachowania czujnika.

W pierwszym podejściu, celem zweryfikowania podstaw zdefiniowanej koncepcji, zdecydowano o opartym na obserwacjach oszacowaniu wartości σ dla każdej z trzech klas [47]. Dłoń ułożoną kolejno w każdą z trzech pód przykładano do czujnika w różnych lokalizacjach i odległościach. Utworzona na potrzeby badań aplikacja komputerowa pokazywała w czasie rzeczywistym na wspólnym wykresie profil mierzony oraz trzy krzywe rozkładu normalnego o zmieniających w trakcie obserwacji wartościach odchylenia standardowego. Jako wielkości charakteryzujące smukłość rozkładu normalnego dla każdej z klas pód dłoni subiektywnie wybrano odpowiednio: $\sigma_{IFS}=0,5$ V, $\sigma_{2FJ}=1$ V oraz $\sigma_{4FJ}=1,5$ V. Podczas badań wykorzystano prototypowy czujnik pracujący w trybie pasywnym, o $n_{PD}=8$, $d_{PD}=1$ cm, $\beta_{PD}=80^\circ$. Wykorzystując podane wartości σ dla każdej z klas przeprowadzono wstępne pomiary dokładności rozpoznawania gestów na grupie 11 ochotników (innych niż w punkcie 4.2.1), z których każdy pokazywał daną pozę trzykrotnie i centralnie nad sensorem. Niewielki zbiór danych i kontrolowane warunki eksperymentu pozwoliły na uzyskanie dokładności klasyfikacji pód zwartych na poziomie 92,42%. Wysoka dokładność wskazała na racjonalność rozwinięcia metody i sprawdzenia jej dla większych zbiorów danych [47].

W drugim podejściu do określenia wartości σ zastosowano obiektywną metodę jej wyznaczania oraz wykonano symulacje z wykorzystaniem opracowanego środowiska bazującego na metodzie Monte Carlo (3.4.2) aby przeprowadzić badania w kontrolowanych warunkach. Postanowiono również wyznaczyć wartości σ z dużą dokładnością toteż w tym celu skonfigurowano wirtualny czujnik o wysokiej rozdzielczości ($n_{PD}=80$, $d_{PD}=1$ mm). Dodatkowo, nie zastosowano w symulacji bloków optycznych, a czujnik emitował fotony równomiernie z całej powierzchni (idealny system podświetlający, tryb aktywny) [154]. W pierwszym zestawie symulacji modelowane kolejno trzy obiekty interakcji fotonu (symulujące trzy typy zwartych pól dłoni), umieszczone środkiem symetrii w punkcie $x_0=0$ cm, $y_0=1$ cm (por. Rys. 3.11). Do każdej z trzech otrzymanych krzywych symulacyjnych dopasowany został rozkład normalny o obiektywnie uzyskanej wartości σ . W tym celu wykorzystano miarę rozbieżności RSS, minimalizując odległość między daną krzywą symulacji, k_S i poszukiwaną krzywą wzorca, k_W :

$$\hat{\omega}(\sigma) = \min_{\sigma} \sum_{i=1}^{n_{PD}} (k_{Si} - k_W(x_i|COG, \sigma))^2 \quad (4.6)$$

gdzie k_{Si} to wartość krzywej symulacji w i -tym punkcie, $k_W(x_i|COG, \sigma)$ to wartość dopasowywanej krzywej rozkładu normalnego w i -tym punkcie o współrzędnej x_i . Dzięki umieszczeniu pól w osi wirtualnego sensora ($x_0=0$ cm) wartość COG symulacji i μ dopasowywanego rozkładu normalnego równa się 0.

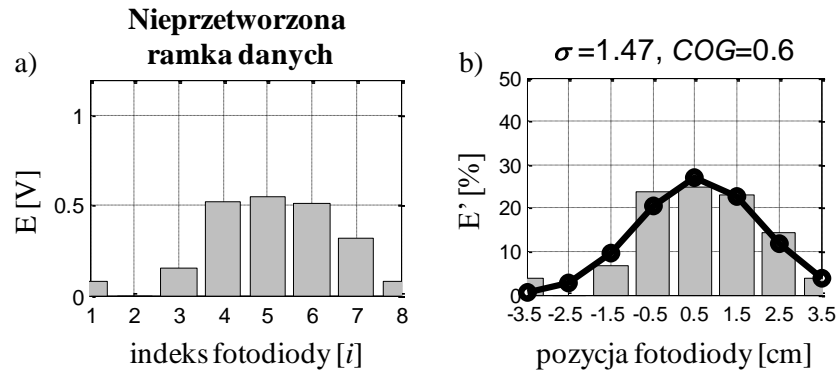
Stosując minimalizację RSS dla każdej krzywej symulacji (stowarzyszonej z inną pozą dłoni) otrzymano 3 wartości σ opisujące odpowiadające im krzywe wzorców (najlepiej dopasowaną do każdej z nich). Dane dotyczące wielkości fizycznych symulowanych pól (obiektów interakcji) i wyniki dopasowania prezentuje Tabela 4.1.

Tabela 4.1 Konfiguracja obiektów interakcji i rezultaty dopasowania wzorca metodą RSS.

l	Poza	R [cm]	w [cm]	σ [V]	RSS(σ)
1	1FS	0,75	0,25	0,754	0,104
2	2FJ	0,75	1,7	1,124	0,094
3	4FJ	0,75	5	2,132	0,212

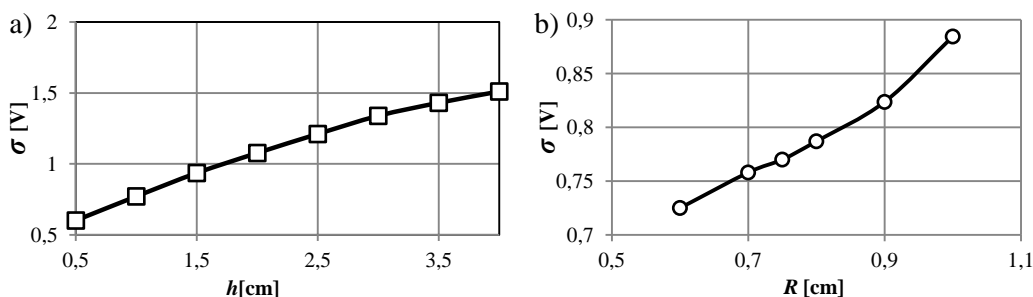
Wykorzystując znaną dzięki wzorowi (4.6) wartość σ i znając wartości pozostałych parametrów (COG , x_i) możliwe jest wykreślenie referencyjnego wzorca rozkładu

normalnego k_W o n_{PD} punktach i przekształconego (znormalizowanego) profilu intensywności światła k_O na wspólnym wykresie (Rys. 4.4).



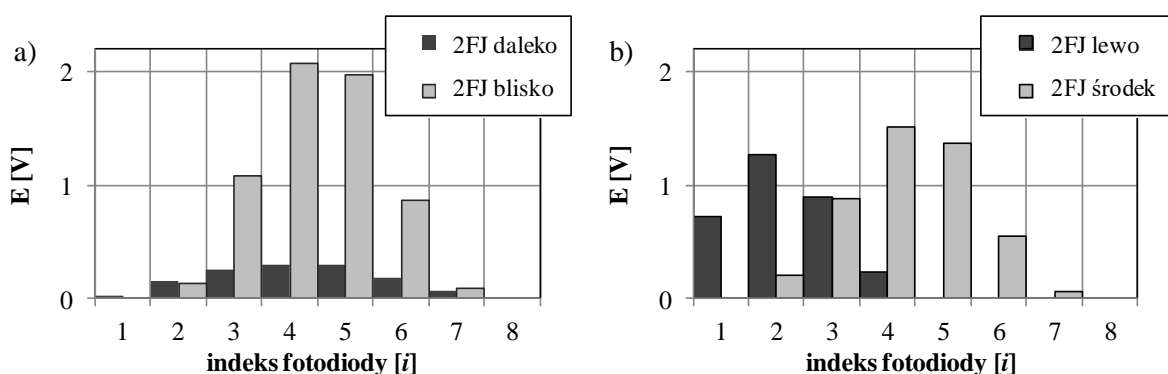
Rys. 4.4 a) Nieprzetworzona ramka danych (wynik pomiaru).
b) Przekształcona ramka danych k_O (wykres słupkowy) i dopasowana krzywa wzorcowa k_W (czarny przebieg).

Należy zaznaczyć, że Tabela 4.1 zawiera wyniki symulacji, w których czujnik był skonfigurowany inaczej niż w podejściu pierwszym do określenia wartości σ opisujących każdą z trzech klas zwartych pól dłoni. W podejściu drugim zdecydowano o zbadaniu wartości σ drogą symulacji, w kontrolowanych warunkach, niezależnych od poziomu światła otoczenia (tryb aktywny). Mimo tego, iż w pierwszym i drugim podejściu rozważano pracę czujnika w różnych trybach (pasywny i aktywny), oraz iż zestawiono wyniki pomiarów i symulacji, otrzymane w podejściu drugim wartości σ dla poszczególnych klas (Tabela 4.1) są zbliżone do intuicyjnie wybranych w podejściu pierwszym. W drugim zestawie symulacji z tego podejścia, zbadano wpływ odległości palca (poza 1FS), h , od sensora na wartość σ dopasowywaną przez minimalizację miary $\hat{w}(\sigma)$ (wzór 4.6) (Rys. 4.5a), która uwidacznia znaczną jego zmienność. Biorąc pod uwagę wartości σ z podejścia pierwszego ($\sigma_{1FS}=0,5$ V, $\sigma_{2FJ}=1$ V, $\sigma_{4FJ}=1,5$ V) i kryterium wzoru (4.5) ten sam palec byłby klasyfikowany jako poza 1FS, 2FJ lub 4FJ w zależności tylko od jego położenia na osi y względem czujnika. Nierównomierności w parametrach spłaszczenia profilu podświetlenia ($flatness_x$ i $flatness_y$) pozwalają przypuszczać, iż również różne położenia poży wzdłuż osi x mogą generować k_W o różnych wartościach σ . Kolejna symulacja pozwoliła także na zaprezentowanie wpływu wielkości palca, R , umieszczonego w ($x_o=0$ cm, $y_o=1$ cm) na σ , który ma jednak wyraźnie mniejsze znaczenie niż zmienność h (Rys. 4.5b).



Rys. 4.5 a) Wpływ odległości palca h na wartość dopasowanej σ (symulacje).
 b) Wpływ szerokości palca R ($x_o=0$ cm, $y_o=1$ cm) na wartość dopasowanej σ (symulacje).

Wykresy na Rys. 4.6 przedstawiają wpływ odległości pozy od czujnika i jej położenia względem środka czujnika na profil intensywności światła odbitego częściowo wyjaśniając zmienność obserwowaną na Rys. 4.5a.



Rys. 4.6 a) Profil pozy 2FJ usytuowanej blisko ($h \sim 1,5$ cm) i daleko ($h \sim 4$ cm) od czujnika gestów.
 b) Profil pozy 2FJ umieszczonej centralnie ($shift \sim 0$ cm) nad i nad lewą ($shift \sim -2$ cm) krawędzią czujnika.

W trzecim podejściu do określenia wartości σ zdecydowano o uwzględnieniu pozycji dłoni względem czujnika w wyznaczaniu wartości poszczególnych σ , stowarzyszonych z daną pozą. Do tego celu wykorzystano zbiór treningowy opisany w poprzednim podrozdziale (4.2.1). Dla każdej obserwacji danej klasy ze zbioru treningowego obliczono parametry COG oraz max (wartość maksymalna z ramki danych) związane z położeniem dłoni odpowiednio w osiach x i y . Znając przesunięcie krzywej obserwacji względem początku układu współrzędnych dzięki COG i wykorzystując wzór (4.6) obliczono stowarzyszoną z daną obserwacją wartość σ . Następnie zdefiniowano model dwóch zmiennych, hipotetycznie przybliżający spodziewaną wartość σ dla danej pozy zwartej o kodzie l , w zależności od położeniem dłoni względem czujnika:

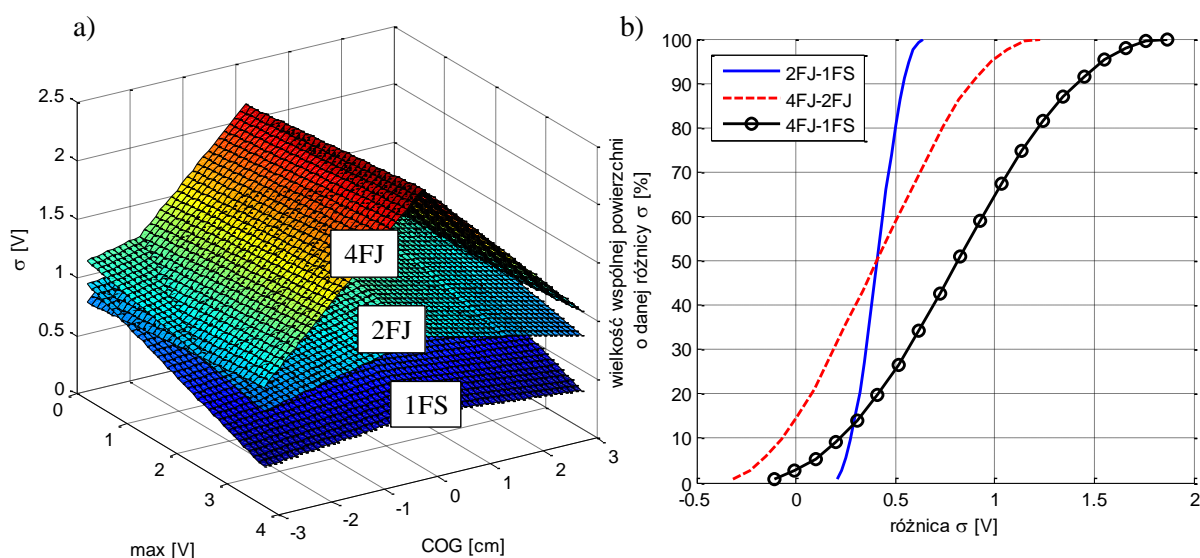
$$\sigma_l(max, COG) = a + b \cdot max + c \cdot abs(COG) \quad (4.7)$$

gdzie a , b oraz c to współczynniki o wartościach poszukiwanych.

Celem znalezienia współczynników wzoru opisującej przewidywaną na podstawie położenia dłoni wartość σ dla każdej pozy z osobna, wykorzystując stowarzyszone ze sobą pary (COG , max) krzywej obserwacji i znalezione σ krzywej wzorca ze wszystkich obserwacji danego gestu, przeprowadzono liniową regresję dwuwymiarową:

$$\begin{aligned}\sigma_{1FS} &= 1,17 - 0,19max - 0,05abs(COG) \\ \sigma_{2FJ} &= 1,52 - 0,11max - 0,10abs(COG) \\ \sigma_{4FJ} &= 2,19 + 0,05max - 0,46abs(COG)\end{aligned}\quad (4.8)$$

Wartości odchylenia standardowego obliczonych regresji wynosiły dla każdej z poz odpowiednio 0,136, 0,18 i 0,213. Posiadając znane wartości współczynników możliwe jest przedstawienie otrzymanych funkcji σ_i (wzór 4.8) w postaci powierzchni na wspólnym wykresie (Rys. 4.7a). Równania rozwiązano dla siatki parametrów o zakresie: $COG <-3; 3>$ cm z krokiem 0,1 cm i $max <0,2; 3,8>$ V z krokiem 0,1 V.



Rys. 4.7 a) Powierzchnie wzorcowych wartości σ dla każdej pozy.
b) Procentowe pokrycie marginesu parametru σ między powierzchniami.

Dla każdej możliwej kombinacji par powierzchni (3 kombinacje: I: 2FJ-1FS, II: 4FJ-2FJ i III: 4FJ-1FS) dla wspólnych punktów siatki obliczono różnicę wartości σ (od powierzchni reprezentującej pozę o większej ilości palców odejmowano powierzchnię opisującą mniejszą ilość palców). Następnie wyrysowano skumulowane histogramy tak obliczonych różnic (Rys. 4.7b). Dla każdej z kombinacji par powierzchni ponad 65% ich wspólnej powierzchni oddalonych jest co najmniej o 0,25 V. Średnie wartości obliczonych różnic dla par I, II, III wynoszą zaś odpowiednio $0,42 \pm 0,09$ V, $0,45 \pm 0,35$ V i $0,87 \pm 0,44$ V

(średnia wartość powierzchni σ dla pól 1FS, 2FJ, 4FJ wynosi odpowiednio $0,73\pm 0,20$ V, $1,15\pm 0,15$ V i $1,60\pm 0,41$ V). Wielkości marginesów pomiędzy poszczególnymi powierzchniami σ_i w stosunku do średniej wartości powierzchni σ_i wskazują zatem na hipotetyczną możliwość rozróżnienia zwartych pól dłoni na podstawie wzorcowych wartości σ dla zarejestrowanego profilu natężenia światła, wyznaczanych ze wzoru 4.8, w znacznym obszarze *strefy detekcji* czujnika optycznego.

Wyniki dokładności detekcji pól zwartych za pomocą niniejszej oraz kolejnej z metod klasyfikacji opisanej w podrozdziale 4.2, sprawdzone na zbiorze testowym, zamieszczone zostały w sekcji 4.2.5 celem bezpośredniego zestawienia właściwości obydwu zaprojektowanych modeli.

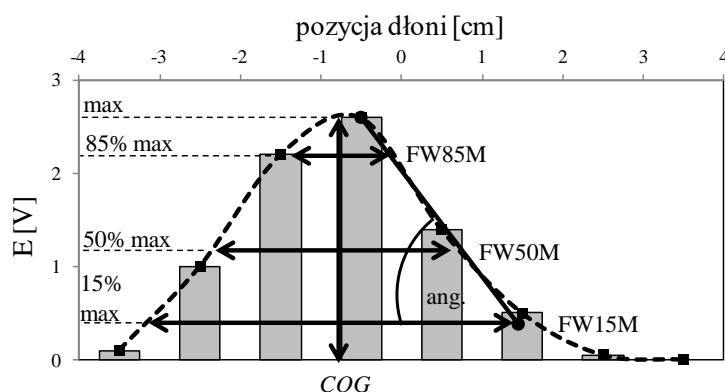
4.2.3. Metoda II: parametryzacja i sztuczne sieci neuronowe

Druga z zastosowanych metod klasyfikacji zwartych pól dłoni, sztuczne sieci neuronowe, jest klasycznym reprezentantem dziedziny uczenia maszynowego popularnie stosowanym w zagadnieniu klasyfikacji. Zdecydowano o wykorzystaniu sparametryzowanych profili natężenia światła (krzywych obserwacji) do wektora cech jako formie danych wejściowych do modelu ANN. W tym celu zaproponowanych zostało 12 cech opisujących krzywą obserwacji. Tabela 4.2 zawiera kod, nazwę i opis każdej z nich.

Tabela 4.2 Rozważane cechy profilu natężenia światła.

Kod	Nazwa	Opis
<i>I</i>	<i>FW15M</i>	Szerokość profilu na poziomie 15% maksimum DF
<i>II</i>	<i>FW50M</i>	Szerokość profilu na poziomie 50% maksimum DF
<i>III</i>	<i>FW85M</i>	Szerokość profilu na poziomie 85% maksimum DF
<i>IV</i>	<i>max</i>	Maksimalna wartość DF
<i>V</i>	<i>COG</i>	Centrum grawitacji DF
<i>VI</i>	<i>mean</i>	Średnia wartość DF
<i>VII</i>	<i>ang</i>	Kąt nachylenia profilu
<i>VIII</i>	<i>sd</i>	Odchylenie standardowe DF
<i>IX</i>	<i>skew</i>	Skośność DF
<i>X</i>	<i>kurt</i>	Kurtoza DF
<i>XI</i>	<i>sm2sd</i>	Ilość wartości w DF mniejszych niż $2 \cdot sd$
<i>XII</i>	<i>grmn</i>	Ilość wartości w DF większych niż <i>mean</i>

Część z cech została przedstawiona w graficznej reprezentacji na Rys. 4.8.

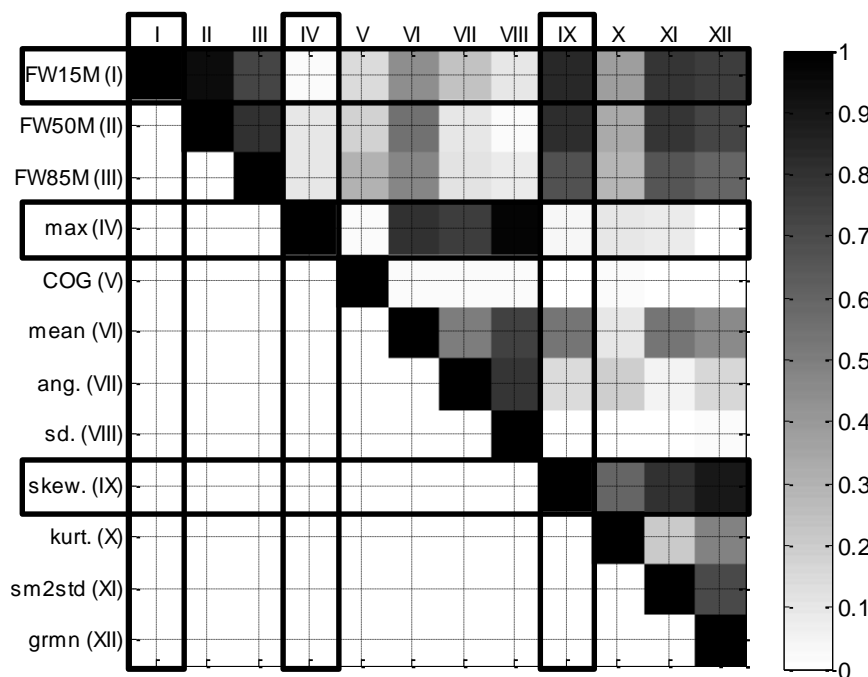


Rys. 4.8 Przykładowy profil natężenia światła rejestrowany przez czujnik wraz z graficzną interpretacją niektórych cech.

W efekcie, analogicznie jak w Metodzie I, dla każdej obserwacji z danej klasy uzyskano zestaw cech (lecz 12 a nie 3) oraz etykietę z nazwą klasy. Istotnym zagadnieniem przed rozpoczęciem procesu uczenia w metodach z zakresu uczenia maszynowego jest właściwy wybór istotnych cech, gdyż najczęściej są one w zróżnicowanym stopniu skorelowane z oczekiwanym wektorem odpowiedzi (etykiet). Większa liczba cech wydłuża proces uczenia lecz niekoniecznie polepsza jego efekty. Zatem niech każda cecha będzie reprezentowana przez wektor o długości równej liczbie obserwacji póż ze wszystkich trzech klas. Dla każdej możliwej pary cech, np. (X, Y) , obliczono współczynnik korelacji Pearsona dla, ρ , zgodnie ze wzorem:

$$\rho_{XY} = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \quad (4.9)$$

gdzie cov oznacza kowariancję wartości cech X i Y a σ to odchylenie standardowe wartości danej cechy. Wartości współczynników korelacji Pearsona dla wszystkich kombinacji cech umieszczono w jednej macierzy (Rys. 4.9). Następnie zdecydowano o wykluczeniu z początkowego zestawu tych cech, które wykazują najwyższą korelację z pozostałymi. Jako nadmiarowe rozważone zostaną każde dwie cechy, dla których odnotowano wartość ρ przekraczającą wartość progową 0,89 (ustaloną na podstawie obserwacji). Usunięta zostanie ta z nich, dla której najwyższa z pozostałych dla danej cechy wartość korelacji ρ z inną cechą jest większa. W rozważanym zestawie warunki takie zostały spełnione przez 3 cechy (Rys. 4.9): I ($FW15M$), IV (max) oraz IX ($skew.$). Podobną metodologię wyboru cech do klasyfikatora, bazującą na współczynniku korelacji zastosował Batchuluun i in. [63].



Rys. 4.9. Macierz absolutnych wartości współczynnika korelacji dla cech. Rzędy i kolumny nadmiarowych cech zostały zaznaczone czarnymi ramkami.

Przy uczeniu sieci neuronowych wzięto zatem pod uwagę 9 cech wejściowych. Postanowiono rozważyć wszystkie możliwe kombinacje obecności danych cech (nie licząc kombinacji w której brak jest jakiegokolwiek cechy), których liczba wynosi $2^9-1=511$. Ilość cech danej kombinacji decyduje o wielkości warstwy wejściowej sieci. Jako model sieci rozważono sieć typu *feedforward* do klasyfikacji wielu klas (posiadającą tyle neuronów w warstwie wyjściowej ile klas w zbiorze treningowym). Wykorzystano metodę poszukiwania siatkowego do wyboru wartości hiperparametrów definiujących najbardziej korzystną topologię sieci. Jako hiperparametry wybrano liczbę warstw ukrytych, liczbę neuronów dla każdej z nich oraz konfigurację cech wejściowych. W pierwszej sesji treningowej rozważone zostały topologie sieci neuronowych o jednej warstwie ukrytej (WU) zawierającej od 2 do 10 neuronów. W drugiej sesji treningowej rozważone zostały topologie o dwóch warstwach ukrytych, również składających się z od 2 do 10 neuronów każda. Biorąc pod uwagę liczbę kombinacji wejściowych cech i liczby neuronów w poszczególnych sesjach, sumarycznie przeprowadzono treningi dla 45990 kombinacji ($511 \cdot 9 + 511 \cdot 9 \cdot 9$). Funkcją aktywacji neuronów w warstwach ukrytych był tangens hiperboliczny (*tanh*), popularnie używana funkcja w sieciach płytkich. Warstwa wyjściowa sieci zawierała zawsze 3 neurony, odpowiadające liczbie klas, których uczony był model. Przed rozpoczęciem procesu uczenia każda cecha zbioru treningowego została przeskalowana tak aby zakres

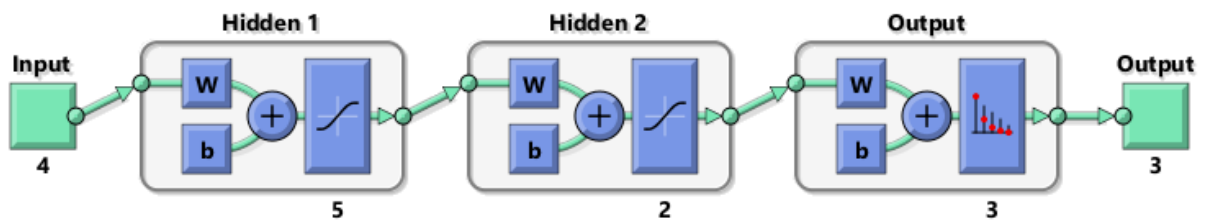
jej wartości mieścił się w przedziale $<-1, 1>$ (ang. *min-max scaling*). Kiedy wartości wszystkich cech leżą w tym samym zakresie sieć neuronowa będzie mogła uczyć się każdej z cech w podobnym tempie.

Do przeprowadzenia uczenia sieci na zbiorze treningowym opisanym w sekcji 4.2.1. wykorzystano funkcję *patternnet* w programie MATLAB®, z ustawieniami: maksymalna ilość epok - 1500, maksymalna liczba błędów - 200 (reszta ustawień, poza hiperparametrami, była domyślna). Uczenie każdej z topologii odbywało się z uwzględnieniem ewaluacji na zbiorze walidacyjnym, na którym sprawdzane były warunki stopu i przerwanie treningu w obiektywnym momencie aby nie dopuścić do nadmiernego dopasowania sieci neuronowej. Po zakończeniu procesu uczenia danej sieci, jej dokładność klasyfikacji na zbiorze walidacyjnym została zapisana jako referencyjna aby na tej podstawie wybrać model, który zostanie sprawdzony na zbiorze testowym. Tabela 4.3 przedstawia po 3 najlepsze topologie sieci z jedną i dwoma warstwami ukrytymi pod względem uzyskanej dokładności klasyfikacji póź na zbiorze walidacyjnym.

Tabela 4.3 Najskuteczniejsze topologie sieci jedno- i dwuwarstwowych w klasyfikacji póź dłoni.

Dokładność [%]	Liczba cech	1. WU	2. WU	Cechy podane jako wejścia
91,77	6	10	-	{II, V, VI, VII, VIII, X}
92,03	7	5	-	{II, III, V, VI, VII, VIII, XI}
92,28	7	5	-	{II, III, V, VI, VII, VIII, X}
92,28	4	5	2	{V, VI, VIII, XI}
93,32	6	8	5	{III, V, VI, VII, VIII, X}
93,01	8	9	5	{II, V, VI, VII, VIII, X, XI, XII}

Spośród otrzymanych topologii należy wybrać jedną, spełniającą założone kryteria, i dopiero wtedy sprawdzić jej dokładność klasyfikacji na nieznanym jej zbiorze testowym. Ze względu na najmniejszą liczbę potrzebnych cech wejściowych, konkurencyjną dokładność klasyfikacji i małą liczbę neuronów (w porównaniu do innych dwuwarstwowych modeli) zdecydowano o wyborze konfiguracji {4+5+2} (liczba cech, liczba neuronów w 1. WU, liczba neuronów w 2. WU)(Rys. 4.10). Mniejsza liczba cech i mniejsza liczba neuronów pozwalają na skrócenie czasu potrzebnego na przygotowanie danych i wykonanie ewaluacji sieci (*forward pass*) co pozwoli na zwiększenie częstotliwości próbkowania czujnika gestów.



Rys. 4.10 Wybrana topologia sieci neuronowej. Schemat wygenerowany przez program MATLAB®.

4.2.4. Pozy rozpostarte

W bloku *Klasyfikacja pozy* zaproponowanego GRS znajdują się osobne bloki odpowiedzialne za rozpoznawanie póż o palcach zwartych (FJ) i rozpostartych (FS). Wejście do każdego z bloków zostało zdeterminowane wartością parametru *PKS*, mówiącego o ilości wykrytych pików (wartości szczytowych) w zarejestrowanym profilu natężenia światła. W każdym profilu wykrywane są wszystkie wartości szczytowe i jeżeli dwa sąsiednie piki dzieli wcięcie o odpowiedniej głębokości, uznaje się, iż pochodzą one od oddzielnych rozpostartych palców. Decyzja o wymaganej głębokości wcięcia określana jest na podstawie wartości parametru I_D (opisanego w sekcji 3.3.4), który musi być większy niż (eksperymentalnie i na podstawie symulacji) dobrana wartość progowa, $I_{DT}=0,15$. Każdy spełniony warunek $I_D > I_{DT}$ zwiększa licznik wartości *PKS* o jedność. Zastosowanie tego kryterium pozwala na ocenę, czy obserwowany profil pochodzi od pozy zwartej ($PKS=1$) czy tworzą go rozpostarte palce w ilości *PKS* i został wykorzystany jako kryterium w bloku decyzyjnym GRS o nazwie *Rodzaj pozy* w fazie *Klasyfikacja pozy*. Funkcjonalność bloku *Klasyfikator FS* sprowadza się do zliczenia wartości *PKS*. Teoretycznie sensor o parzystej liczbie n_{PD} fotodiod pozwala na rozróżnienie do $n_{PD}/2$ palców (na przemian szczyty i doliny w profilu natężenia światła). Zatem czujnik o 8 fotodiodach umożliwiać może rozróżnienie do 4 rozpostartych palców jednak jest to możliwe jedynie dla małej ilości położeń dłoni w stosunku do sensora. Z tego względu i z przyczyn fizjonomicznej wygody pokazywania poszczególnych póż zdecydowano, iż z póż z rozpostartymi palcami rozważane będą jedynie konfiguracje 1FS i 2FS.

4.2.5. Porównanie metod klasyfikacji póż zwartych

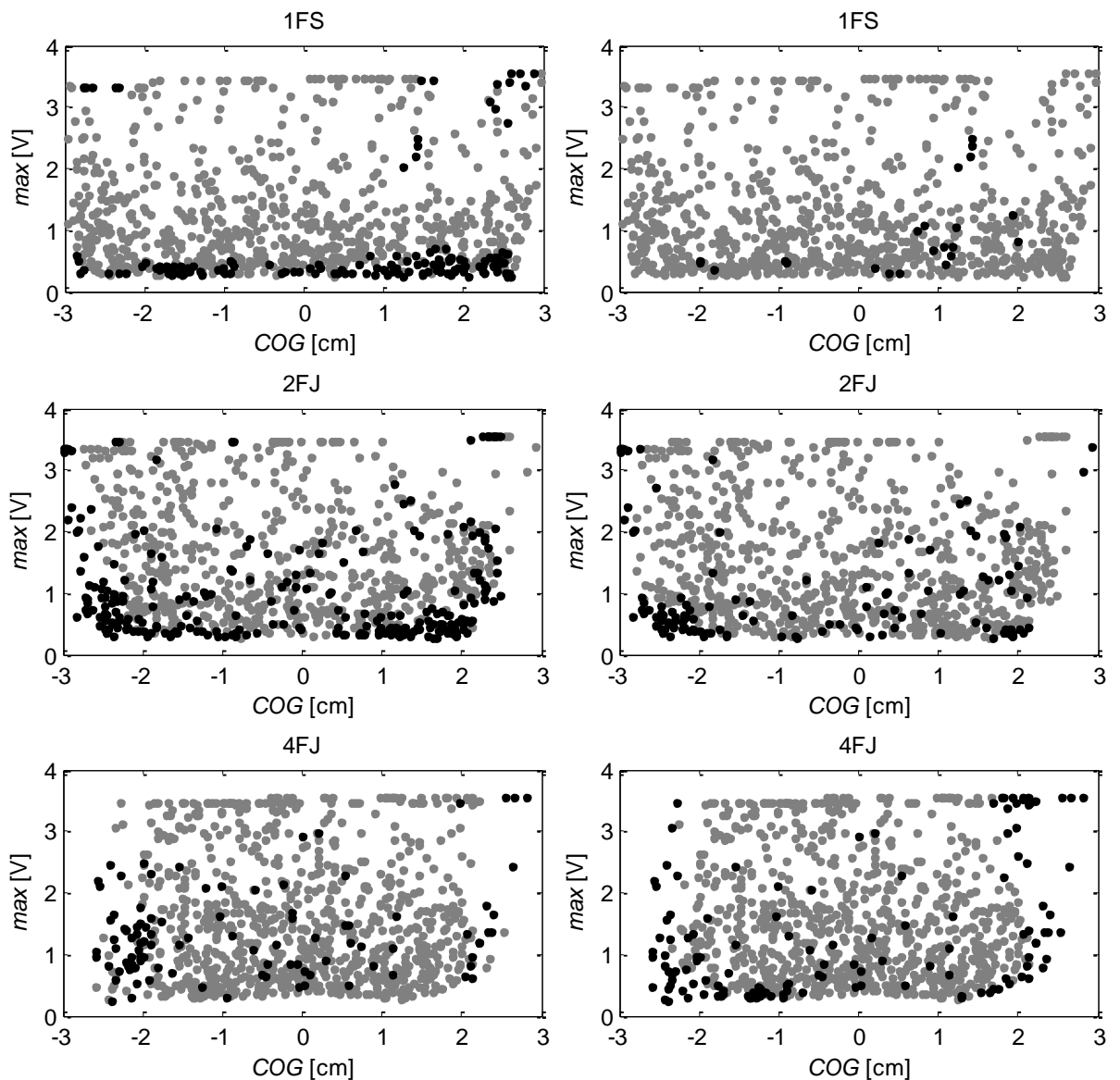
Wykorzystując zbiór testowy porównano dokładność rozpoznawania póż dłoni przez obydwie wykorzystane metody. Uśredniona po wszystkich trzech pozach dłoni dokładność klasyfikacji dla Metody I wynosi 81,19%, zaś dla Metody II 90,02%. Wynik ewaluacji Metody II na zbiorze testowym jest wyraźnie niższy od tego ze zbioru walidacyjnego

(92,28%) co może być spowodowane przeuczeniem sieci czy jej ograniczoną pojemnością (mała liczba neuronów, która była jednak pożądana ze względu na szybkie działanie sieci w mikrokontrolerze). W następnej kolejności, wykorzystując zbiory testowe, obliczono tablice pomyłek pozwalające na obserwację dokładności klasyfikacji poszczególnych póż oraz tego, w jakim stopniu zaproponowane modele myliły je z innymi pozami (Rys. 4.11).

		Metoda I Dopasowanie krzywych			Metoda II Sieci neuronowe				
		Przewidywane			Przewidywane				
		a)	1FS	2FJ	4FJ	b)	1FS	2FJ	4FJ
Prawdziwe	1FS		83,19	12,51	4,30		97,85	1,91	0,24
	2FJ		9,02	71,05	19,93		6,88	87,19	5,93
	4FJ		1,19	9,49	89,32		1,51	13,48	85,01

Rys. 4.11 Tablice pomyłek dla dwóch metod rozpoznawania póż dłoni wykonanych na zbiorze testowym.
a) Metoda I - dopasowanie krzywych wzorca i obserwacji. b) Metoda II - sztuczne sieci neuronowe.

Następnie w postaci map detekcji wykreślone zostały lokalizacje prawidłowo i niewłaściwie rozpoznanych póż dłoni wykorzystując parametry *COG* i *max* jako wartości związane z położeniem dłoni w relacji do czujnika (Rys. 4.12).



Rys. 4.12 Mapy prawidłowo (szare kropki) i błędnie (czarne kropki) zaklasyfikowanych póz względem parametrów stowarzyszonych z zarejestrowaną pozycją dłoni (COG i max).

Zaprezentowane wyniki pokazują, iż klasyfikator bazujący na sztucznych sieciach neuronowych charakteryzuje się znacznie wyższą dokładnością rozpoznawania póz dłoni (niemal 9% wyższa dokładność) niż klasyfikator bazujący na dopasowaniu krzywych wzorca i obserwacji. Tablice pomyłek pokazują jednak, że poza 4FJ jako jedyną zwartą pozę jest lepiej rozpoznawana przez Metodę II (mimo znacznego nakładania się płaszczyzny σ_{4FJ} na pozostałe). Metoda I wykazuje tendencję do niedoszacowania smukłości kształtu obserwacji - błędy detekcji zarówno dla pozy 1FS jak i 2FJ są zdominowane przez wskazania na pozę o większej ilości palców. W Metodzie II nie dopatrzonego się tendencji podobnego typu. Z map detekcji wynika, iż większość błędnie zaklasyfikowanych póz

była zlokalizowana na skraju pola widzenia czujnika ($\text{abs}(COG) > 2$) i/lub w pobliżu górnej granicy jego *strefy detekcji* (mała wartość *max*). Z tego względu, zaproponować można dodatkowe ograniczenie dla dopuszczalnej lokalizacji pozy, tak aby generowała ona profil o wartości *max* co najmniej na poziomie 0,8 V. W takim wypadku dokładność detekcji na obserwacjach spełniających ten warunek wzrosła do 91,72%.

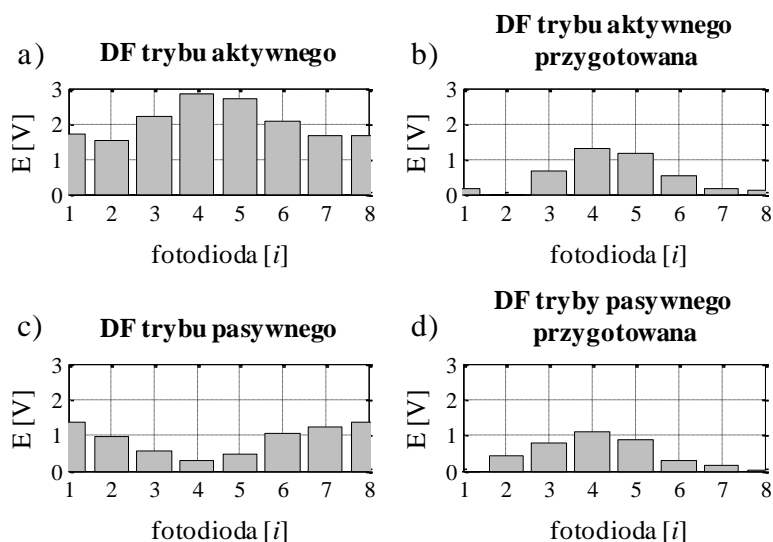
W kolejnym kroku zestawienia wydajności obydwu metod każdy z modeli został zaimplementowany w języku C i przeniesiony do pamięci mikrokontrolera czujnika optycznego. W ramach ewaluacji pełnej funkcjonalności Metody I dla każdej ramki danych należało zaimplementować obliczanie wartości parametrów *COG* i *max* oraz 3 równań na wartość σ_l dla każdej z 3 pól, obliczyć po 8 wartości 3 rozkładów k_m z otrzymanymi σ_l każdy i dla każdego z nich i pomierzonej krzywej k_o miarę rozbieżności (wzór 4.4). Metoda II wymagała implementacji obliczenia 4 cech (*COG*, *mean*, *sd*, *sm2std*) oraz zrealizowania funkcjonalności dwuwarstwowej sieci neuronowej z wagami i obciążeniami wyeksportowanymi ze środowiska MATLAB®. Celem uproszczenia i oszczędności czasu obliczeń, funkcje aktywacji implementowano jedynie w warstwach ukrytych i zrezygnowano z funkcji *softmax* warstwy wyjściowej na rzecz funkcji linowej. Jako odpowiedź sieci pobierano numer neuronu o największej wartości w warstwie wyjściowej. Za pomocą oscyloskopu pomierzony został czas potrzebny mikrokontrolerowi na wykonanie pełnego cyklu klasyfikację ramki danych (przygotowanie cech, ewaluacja modelu) przez każdą z metod. Dla pomierzonych czasów (t) wyznaczono okresy ($1/t$), które zaokrąglono w górę do najbliższej całkowitej wielokrotności 5Hz. Dla Metody I uzyskano częstotliwość 10Hz zaś Metoda II pozwoliła na rozpoznawanie pól dłoni w pięciokrotnie krótszym czasie (50Hz). Główną przyczyną tego stanu rzeczy jest konieczność wykonywania kosztownych czasowo dla mikrokontrolera operacji podnoszenia liczb zmiennoprzecinkowych do potęgi przy obliczaniu rozkładów normalnych.

Na podstawie wyższej dokładności klasyfikacji pól i szybszego działania, wybrano Metodę II, dwuwarstwowy model sztucznej sieci neuronowej, jako rozpatrywaną w pierwszej kolejności w dalszych badaniach związanych z rozpoznawaniem gestów. Została ona również przypisana jako funkcjonalna realizacja bloku *Klasyfikator FJ* w fazie *Klasyfikacja pozy* systemu rozpoznawania gestów.

4.3. Zachowanie sensora w różnych warunkach oświetlenia

W dotychczasowo opisanych badaniach rozważano głównie zdolność sensora do klasyfikacji póż dłoni podczas jego pracy w trybie aktywnym, przy pomijalnej obecności światła otoczenia. Następnym krokiem w ewaluacji czujnika liniowego jest zweryfikowanie jego użyteczności w trybie pasywnym, w różnych warunkach oświetlenia naturalnego.

W odróżnieniu od aktywnego, w pasywnym trybie pracy wykorzystywane jest wyłącznie światło zewnętrzne, bez dodatkowego źródła światła (jak diody LED). W tym trybie światło będzie blokowane ręką wykonującą gest wytwarzając wklęsły profil intensywności światła (a więc profil stopnia zacienienia) na powierzchni sensora. Zauważono, iż po odwróceniu spróbkowanego profilu jego kształt podobny jest do wzoru natężenia światła uzyskanego w trybie aktywnym. Podobieństwo zaobserwowano po zastosowaniu następującej procedury przygotowania obydwu pakietów. Dla trybu aktywnego - odjęcie od otrzymanej DF jej wartości $\min(DF)$ (tak aby zakres wartości zaczynał się zawsze od 0). Dla trybu pasywnego zaś - inwersja (pomnożenie przez -1) i dodanie $\min(DF)$ z DF po inwersji (dzięki czemu również i zakres tego profilu będzie zaczynał się od 0). Rys. 4.13a, c prezentują profile intensywności światła zarejestrowane w podobnych warunkach oświetlenia (pochmurny dzień) dla pozy 2FJ ale w różnych trybach pracy. Rys. 4.13b, d pokazują efekt przygotowania ramek danych trybu aktywnego i pasywnego uwidaczniając podobieństwo kształtów.



Rys. 4.13 Profil intensywności światła spróbkowany w trakcie pochmurnego dnia wywołany pozą 2FJ umieszczoną przed czujnikiem działającym w różnych trybach pracy. a) DF trybu aktywnego. b) Przygotowana DF trybu aktywnego. c) DF trybu pasywnego. d) Przygotowana DF trybu pasywnego.

Pytaniem jest, czy operacja przygotowania profilu zarejestrowanego w trybie pasywnym pozwoli czujnikowi gestów zachować swoje właściwości rozpoznawania gestów (określanie fazy nasłuchiwanie gestu, rozpoznawanie pozy, śledzenie lokalizacji dłoni) i jak zależeć będą one od różnych warunków oświetlenia otoczenia. Szczegółowy opis i pełna zawartość przeprowadzonych w związku z tymi badaniami eksperymentów znajdują się w jednej z publikacji autora niniejszej rozprawy [158]. Poniżej przedstawiono zaś tę część badań i wyników zawartych w publikacji, która odnosi się do tez postawionych w niniejszej rozprawie doktorskiej.

4.3.1. Walidacja czujnika liniowego w trybie pasywnym

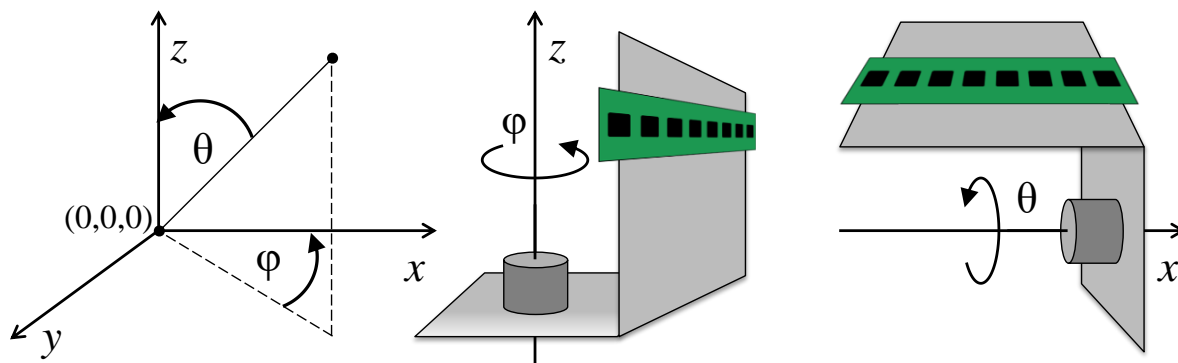
Źródło światła otoczenia może być naturalne (światło słoneczne) lub sztuczne (np. światło żarowe stosowane w pomieszczeniach). W niniejszych badaniach skupiono się na analizie wpływu światła naturalnego. Poziom natężenia światła otoczenia, jego kierunkowość i kąt padania na czujnik mogą mieć wpływ na użyteczność i niezawodność pasywnego trybu działania optycznego czujnika gestów o liniowej budowie. Dlatego celem eksperymentów opisanych w tym podrozdziale jest zbadanie wpływu świetlnych warunków otoczenia na parametry wyjściowe czujnika gestów pracującego w trybie pasywnym.

Jak podano wcześniej, nie wszystkie rejestrowane przez sensor ramki danych powinny być przekazywane do dalszych bloków funkcjonalnych GRS w celu klasyfikacji pozy. W trybie aktywnym, jedynie gdy wartość $sd(DF)$ przekracza ustaloną wartość progową ($sd_{Ta}=0,1$ V), uruchamiane są bloki analizy zarejestrowanego profilu. Jednak odpowiednia wartość progowa dla trybu pasywnego może być inna, co należy zbadać. Ponadto celem eksperymentów było również oszacowanie warunków świetlnych, dla których należy zmienić tryb pracy czujnika optycznego. Dlatego w trakcie niniejszych badań przeprowadzono wstępne przetwarzanie wszystkich ramek danych w celu obliczenia parametrów $sd(DF)$, $max(DF)$, $min(DF)$ i ich różnicy ($diff=max(DF)-min(DF)$), tak aby rozważyć kryteria różnych parametrów dla przełączania trybów pracy.

Innymi parametrami obliczonymi na podstawie ramek danych, które obserwowano podczas testów użyteczności czujnika w różnych warunkach oświetlenia były kod rozpoznanej pozy dłoni i składowa wzdłużna lokalizacji ręki w stosunku do czujnika. Kod reprezentujący pozę ręki był wynikiem działania wybranego klasyfikatora zwartych pól dłoni (Metoda II – ANN). Drugi parametr, lokalizacja ręki była szacowana poprzez obliczenie

środka ciężkości (*COG*) profilu intensywności światła (zacienienia) reprezentowanego przez wartości DF po przygotowaniu ramki danych.

Aby ocenić dopuszczalne dla trybu pasywnego czujnika gestów warunki światła otoczenia, przeprowadzono eksperymenty zgodnie z następującą procedurą. W pierwszej kolejności czujnik optyczny zamontowano na stole i obracano w osiach φ i θ (sferyczny układ współrzędnych) tak (Rys. 4.14), aby zmierzyć charakterystykę referencyjną światła otoczenia (parametr $max(DF)$) w pomieszczeniu, w którym odbywały się pomiary. Następnie przed czujnikiem montowano w dodatkowym uchwycie przeszkodę naśladującą wybraną pozę dłoni. Przeszkoda była przymocowana do głównego uchwytu na czujnik tak, że obroty czujnika względem osi nie zmieniały wzajemnego położenia czujnika i przeszkody. Następnie wykonywano pomiary rotacyjne w obydwu osiach, a otrzymane wyniki normalizowano za pomocą charakterystyk referencyjnych (dzielono otrzymany przebieg przez referencyjny). Każda zmiana poziomu oświetlenia otoczenia rejestrowana była przez osobny światłomierz. Hipotetyczna korelacja wynikowej znormalizowanej charakterystyki ze zmiennością poziomu intensywności światła z zewnętrznego światłomierza oznaczałaby, że parametr $max(DF)$ (przed przygotowaniem ramki danych) zmienia się wraz ze światłem otaczającym, bez względu na pozycję czujnika w odniesieniu do kierunku słońca. Jako przeszkodę wybrano pozę 2FJ gdyż wytwarza ona profil zacienienia o umiarkowanej szerokości (w stosunku do szerokości zastosowanego czujnika). Jej cień najprawdopodobniej nie pokryje całej szerokości czujnika, więc uzyskana wartość parametru $max(DF)$ będzie zbliżona do wartości maksymalnej, która byłaby rejestrowana bez obecności jakiegokolwiek przeszkody umieszczonej naprzeciwko czujnika. Podczas obrotów, w określonych pozycjach (kątach) względem źródła światła (słońca), w których klasyfikator prawidłowo rozpozna pozę, warunki światła otoczenia (reprezentowane przez parametr $max(DF)$) zostaną uznane za dopuszczalne. Błędne rozpoznanie pozę będzie interpretowane jako przeprowadzone w niewystarczająco jasnych dla trybu pasywnego warunkach.



Rys. 4.14 Schemat relacji między osiami obrotu φ i θ a pozycją czujnika liniowego w eksperymentach z obrotami czujnika. Szary element to uchwyt obrotowy, do którego mocowano czujnik optyczny.

4.3.2. Baza póż dłoni dla dwóch trybów pracy i klasyfikacja póż

W pierwszej części badań skompletowano bazę zwartych póż dłoni rejestrowanych przez czujnik w różnych trybach pracy i w różnych warunkach oświetlenia. Jako zbiór danych trybu aktywnego wykorzystano zbiór opisany w punkcie 4.2.1, rejestrowany przy braku obecności światła zewnętrznego. Zbiór danych trybu pasywnego zarejestrowany został w czterech sesjach zróżnicowanych pod względem warunków oświetlenia zewnętrznego, możliwie liniowo rozłożonych ze względu na poziom jasności. Dzięki przeprowadzaniu badań w różne dni i o różnych porach dnia średnia jasność podczas kolejnych sesji zbierania danych wynosiła odpowiednio 230, 700, 1460 i 2200 luksów. Jasność rejestrowana była za pomocą miernika Lutron LX-105. Dla każdej sesji pomiarowej uzyskano zbiór o całkowitej liczbie 1500 póż. Zostały one zgromadzone przy udziale trzech ochotników (średni wiek 29 lat, mężczyźni), którzy wygenerowali po 500 pakietów dla wszystkich trzech póż zwartych dla każdej z sesji zbierania danych. Liczba osób biorących udział w tym eksperymencie nie była kluczowa gdyż skupiono się na badaniu wpływu zróżnicowanych warunków świetlnych na właściwości czujnika optycznego. W sumie uzyskano zatem po 2000 obserwacji każdej z trzech póż, w 4 różnych warunkach oświetlenia. Procedura rejestracji póż była podobna jak w przypadku zbioru trybu aktywnego, lecz bez osłony czujnika przed światłem. Rejestrację póż przez czujnik pracujący w trybie pasywnym przeprowadzono z powierzchnią czujnika umieszczoną prostopadle do kierunku źródła światła (warunki sprzyjające – prostopadle do płaszczyzny podłogi i z fotodiodami ukierunkowanymi w stronę okien pomieszczenia).

Obydwa zbiory danych zostały podzielone na podzbiory treningowe, walidacyjne i testowe o proporcjach 70%/15%/15%. Celem poglądowego rozpoznania dokładności klasyfikacji póż na podstawie danych zarejestrowanych w różnych trybach pracy

zdecydowano o wytrenowaniu osobnych klasyfikatorów dla każdego z trybów czujnika. Na podstawie ewaluacji klasyfikatorów póż (4.2.5) zdecydowano o treningu sztucznych sieci neuronowych w programie MATLAB®. Wytrenowane zostały dwa modele sieci neuronowych. Sieć o nazwie *aANN* (ANN dla trybu aktywnego) została wytrenowana z użyciem zbioru treningowego danych zgromadzonych w trybie aktywnym. Analogicznie, *pANN* (ANN dla trybu pasywnego) wytrenowano stosując zbiór treningowy zarejestrowany w trybie pasywnym.

W tym podejściu minimalizacja rozmiaru sieci neuronowej i szybkości działania metody nie była najważniejszym czynnikiem. Dlatego zdecydowano o wytrenowaniu nowego, w stosunku do modelu z sekcji 4.2.3, klasyfikatora – *aANN* – dla danych trybu aktywnego, posiadającego potencjalnie większe zdolności do rozpoznawania póż dzięki zwiększeniu liczby neuronów sieci i cech wejściowych.

W tym badaniu, przy treningu obydwu modeli wzięto pod uwagę tylko topologie jednowarstwowe ale z ilością od 1 do 30 neuronów (w poprzednich badaniach limit wynosił 10) w warstwie ukrytej. Ponadto, zdecydowano także o zwiększeniu liczby wejść do sieci do 9. Wobec tego, dla sieci trenowanej na danych z trybu aktywnego jako wejścia podano wszystkie cechy, które nie zostały wyeliminowane na podstawie macierzy współczynników korelacji zdefiniowanej w punkcie 4.2.3: *FW50M*, *FW85M*, *COG*, *mean*, *ang*, *sd*, kurtoza, liczba wartości w DF poniżej $2 \cdot sd$ oraz liczba wartości w DF większych od średniej. W efekcie, w ramach poszukiwania siatkowego optymalnej topologii sieci neuronowej, rozważono jeden hiperparametr (ilość neuronów warstwy ukrytej). Dla takiej konfiguracji wejść, najbardziej wydajna topologia sieci składała się z 22 neuronów w warstwie ukrytej. Uzyskana dokładność klasyfikacji póż przez model *aANN* na zbiorze testowym wyniosła 93,46% w porównaniu do 90,02% uzyskanych wcześniej w 4.2.5. Następnie, model *aANN* przetestowano na zbiorze danych testowych trybu pasywnego a uzyskana dokładność klasyfikacji była niższa niemal o 20p.p. (75,51%).

Zestaw cech profilu trybu pasywnego został rozszerzony o parametr *rawmax*, który opisuje maksimum uzyskanego profilu intensywności światła jeszcze przed przygotowaniem ramki danych (inwersja i przesunięcie zakresu). Wykorzystując tę samą metodę eliminacji (macierz współczynników korelacji) również wybrano 9 najważniejszych cech jako wejścia do sieci neuronowej. Wybrane cechy to: *FW15M*, *FW85M*, *COG*, *mean*, *ang*, skośność, kurtoza, liczba wartości w DF mniejsza niż $2 \cdot sd$ i *rawmax*.

Pozostałe ustawienia były takie same, jak w przypadku uczenia sieci *aANN*. Najbardziej dokładna topologia z trenowanych na danych pasywnych posiadała 25 neuronów w warstwie ukrytej i uzyskała dokładność klasyfikacji zbioru testowego na poziomie 98,76%. Dokładność *pANN* została również oceniona na zestawie testowym trybu aktywnego, uzyskując jednak zaledwie 52,82% poprawnej klasyfikacji póż. Sprawdzanie krzyżowe (ang. *cross-checking*) zostało przeprowadzone ponieważ profile aktywne / pasywne po normalizacji były podobne kształtem. Podsumowanie wyników przedstawia Tabela 4.4.

Tabela 4.4 Dokładność klasyfikatorów wytrenowanych na danych trybu aktywnego (*aANN*) i pasywnego (*pANN*) uzyskana ze zbiorów testowych poszczególnych trybów pracy czujnika.

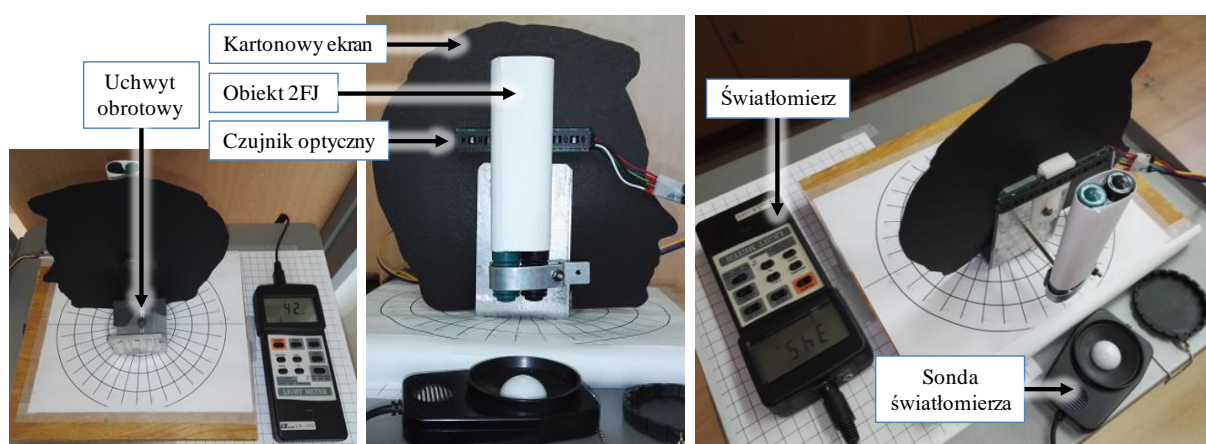
			Typ zbioru testowego		
			<i>Pasywny</i>	<i>Aktywny</i>	
Klasyfikator	<i>pANN</i>	Typ zbioru treningowego	<i>Pasywny</i>	98,76%	52,82%
	<i>aANN</i>	<i>Aktywny</i>	75,51%	93,46%	

Zaprezentowane wyniki wskazują jednoznacznie, iż mimo wizualnego podobieństwa profili aktywnego i pasywnego po transformacji sensor operujący w trybie pasywnym wykazuje znacznie wyższą dokładność rozpoznawania póż kiedy korzysta z klasyfikatora wytrenowanego na danych również z trybu pasywnego. Z tego powodu, w dalszej części badań związanych z wpływem zmiennego poziomu światła otoczenia na właściwości czujnika stosowany będzie klasyfikator póż *pANN*.

4.3.3. Układ pomiarowy z platformą obrotową

Układ pomiarowy wykorzystany w badaniach został zaprojektowany w taki sposób, aby umożliwić oszacowanie wydajności czujnika w różnych warunkach oświetlenia. Sam bowiem obrót czujnika w stosunku do kierunku słońca zmienia kąt padania światła słonecznego na czujnik, a zatem także warunki pomiaru. Dlatego pomiary przeprowadzono z wykorzystaniem skonstruowanej platformy z obrotowym uchwytem. Zgodnie ze sferycznym układem współrzędnych, umożliwia ona rotacje czujnika w kątach φ i θ , jak pokazano to na Rys. 4.14.

Czujnik gestów przymocowany do obrotowego uchwytu, w pozycji gotowej do przeprowadzenia eksperymentów z rotacją w kącie φ przedstawiono na Rys. 4.15. Teksturowy ekran o kształcie ludzkiej głowy widzianej z profilu przymocowano do tylnej części uchwytu na czujnik tak, iż jego pozycja względem czujnika pozostawała stała w trakcie obrotów. Dodanie profilu głowy miało na celu odtworzenie scenariusza zastosowania gdzie czujnik wbudowany jest w ramkę multimedialnych okularów. W przeprowadzonych doświadczeniach obrotowy uchwyt z czujnikiem zamontowany był na platformie na stoliku, 1 m nad ziemią. Półkulistą sondę światłomierza umieszczono na tym samym stole, częścią rejestrującą do góry.



Rys. 4.15 Opis konfiguracji laboratoryjnej. Stanowiąca przeszkodę imitacja pozy 2FJ zamontowana jest przed czujnikiem.

Podczas eksperymentów stolik z platformą i czujnikiem umieszczany był na środku pokoju, w pozycji początkowej czujnika zwróconej w stronę jedynego okna (strona południowa) i z dłuższym bokiem urządzenia równoległe do ziemi. W celu częściowej stymulacji różnych warunków oświetlenia obracano cały układ czujnika na platformie w danej osi o określony kąt. W każdym z następujących po sobie położeniach (kątach) czujnika względem źródła światła zapisywano ramkę danych DF i przesyłano ją do komputera PC za pomocą interfejsu UART czujnika. Poziom oświetlenia otoczenia monitorowany za pomocą miernika Lutron także zapisywano na komputerze PC przy użyciu interfejsu RS232.

Jak opisano wcześniej, do uchwytu obrotowego przymocować można zdejmowaną przeszkodę, naśladującą pozę złączonych dwóch palców (2FJ). Jej użycie zostało wykorzystane w celu obserwacji wpływu zróżnicowanych warunków oświetlenia otoczenia na parametry DF stabilnego układu czujnik-poza dłoni. Szerokość sztucznych palców ($D_2=3,2$ cm) została zaczerpnięta z badania przeprowadzonego na grupie ochotników

(Tabela 3.2). Imitację palców stanowiły dwa walce o $R=0,8$ cm, owinięte białą kartką (efektywne parametry czujnika w nomenklaturze obiektów interakcji symulatora to $w=1,6$ cm, $R=0,8$ cm). Przeszkoda montowana była naprzeciwko środka czujnika, w odległości około połowy zakładanej wysokości *strefy detekcji* ($x_o=0$ cm, $y_o=2$ cm) [157].

W eksperymentach ze zmianą wartości kąta φ pomiędzy czujnikiem a kierunkiem światła, uchwyt obrotowy z czujnikiem obracany był zgodnie z ruchem wskazówek zegara, z krokiem 10° , wykonując pełne koło (36 pozycji). W eksperymentach zmiany kąta θ uchwyt był obracany początkowo w kierunku fotodiodami do sufitu, a następnie do tyłu (względem pozycji wyjściowej), także z krokiem 10° , wykonując półokrąg (19 pozycji). Każdy z eksperymentów kątowych przeprowadzono w 4 scenariuszach. W pierwszych dwóch czujnik mierzył charakterystykę pomieszczenia - pomiar „bez przeszkód” - dla słabego (100 - 400 luksów) i mocnego (600 - 2000 luksów) światła zewnętrznego. Zweryfikowano, iż połowa poziomu nasycenia fotodiod zastosowanych w czujniku (1,9 V) odpowiada poziomowi 592 luksów dla światła słonecznego (pomierzone miernikiem Lutron), a więc jest ona na granicy między zdefiniowanymi warunkami światła otoczenia. W trzecim scenariuszu światło z otoczenia mieściło się w zakresie słabego światła, ale przed czujnikiem została umieszczona przeszkoda 2FJ. W czwartym scenariuszu przeszkoda była również obecna, lecz poziom światła otoczenia mieścił się w mocniejszym zakresie.

Stosunek składowych światła kierunkowego i rozproszonego może mieć wpływ na wygląd profilu cienia wytwarzanego przez przeszkodę, wpływając w ten sposób na parametry ramki danych obliczane przez czujnik. Zjawisko to może być bardziej uwidocznione przy zmianie odległości między przeszkodą a czujnikiem. Dlatego, dla zróżnicowanych warunków oświetlenia zewnętrznego, przeprowadzono również pomiary ze sztuczną przeszkodą 2FJ umieszczoną początkowo przed czujnikiem w odległości $h=1$ cm i przesuwaną aż do $h=10$ cm, z krokiem 1 cm. Pomiary przeprowadzono w trzech scenariuszach, wszystkie z zamieszczoną przeszkodą 2FJ. Poza słabymi i mocnymi warunkami świetlnymi opisanymi w poprzednim akapicie, dodano również scenariusz pomijalnego światła (poniżej 100 luksów). Ponieważ zagadnienia te odnoszą się pośrednio do tez postawionych w niniejszej rozprawie, bardziej szczegółowy opis eksperymentów, wyników i dyskusji dotyczących tej części badań nie został zamieszczony

w rozprawie doktorskiej (jest on natomiast dostępny w jednej z publikacji autora niniejszej rozprawy [158]).

4.3.4. Charakterystyka czujnika liniowego względem warunków oświetlenia

Jednym z głównych zadań GRS obsługującego czujnik optyczny było podjęcie decyzji czy otrzymana ramka danych niesie ze sobą wystarczającą informację aby podjąć proces rozpoznawania pozy dłoni, czy też należy pominąć ją w analizie. W poprzednich badaniach, przy aktywnym trybie pracy i przy braku oświetlenia otoczenia próg wartości odchylenia standardowego sd_{Ta} (blok decyzyjny *Czujnik widzi dłoń*) został ustawiony na 0,1 V (rozdział 4.1). Niemniej jednak wartość ta może nie być adekwatna dla trybu pasywnego czyli w obecności światła otoczenia, mimo, iż kalibracja czułości fotodiod w warunkach laboratoryjnych niweluje wzrost odchylenia standardowego ramki danych ze wzrostem poziomu natężenia światła otoczenia (rozdział 3.5.6). Celem sprawdzenia tej zależności w warunkach rzeczywistych (niekontrolowanych) dla czterech pomiarów kątowych „bez przeszkód” przed czujnikiem (obroty w kątach φ , θ w obecności słabego i mocnego światła) obliczono wartości $sd(DF)$ w 110 pozycjach ($2 \cdot 36 + 2 \cdot 19$). Wartość natężenia światła otoczenia mieściła się w zakresie od 229 do 1108 luksów. Uzyskana rozpiętość wartości $sd(DF)$ wynosiła od 0,0052 do 0,32 V. Ustawienie wartości progu sd_{Tp} (pasywny) na wartość taką jak sd_{Ta} czyli 0,1 V spowoduje, iż 82,72% ramek danych zostanie poprawnie sklasyfikowanych jako nie widzące przeszkody (19 ramek danych miało $sd(DF) > sd_{Tp}$). Podniesienie sd_{Tp} do wartości 0,13 V zwiększyło dokładność klasyfikacji do 94,54%. Dalsze podnoszenie wartości progu powoduje większe ograniczenie zakresu pracy sensora co jest niepożądanym efektem. Mimo kalibracji czułości fotodiod część ramek danych „bez przeszkód” wykazała się własnością $sd(DF) > sd_{Tp}$ co mogło być spowodowane niedokładnością w umieszczeniu bloku optycznego względem fotodiod oraz odbiciami silnego światła od powierzchni różnego typu obecnych w pomieszczeniu laboratoryjnym. W analizie eksperymentów związanych z pasywnym trybem pracy zastosowano zatem wartość progową $sd_{Tp} = 0,13$ V. Bardziej precyzyjne dostosowanie tej wartości może zostać zbadane w trakcie przyszłych badań oraz powinno być uwzględnione w procedurze kalibracyjnej czujnika.

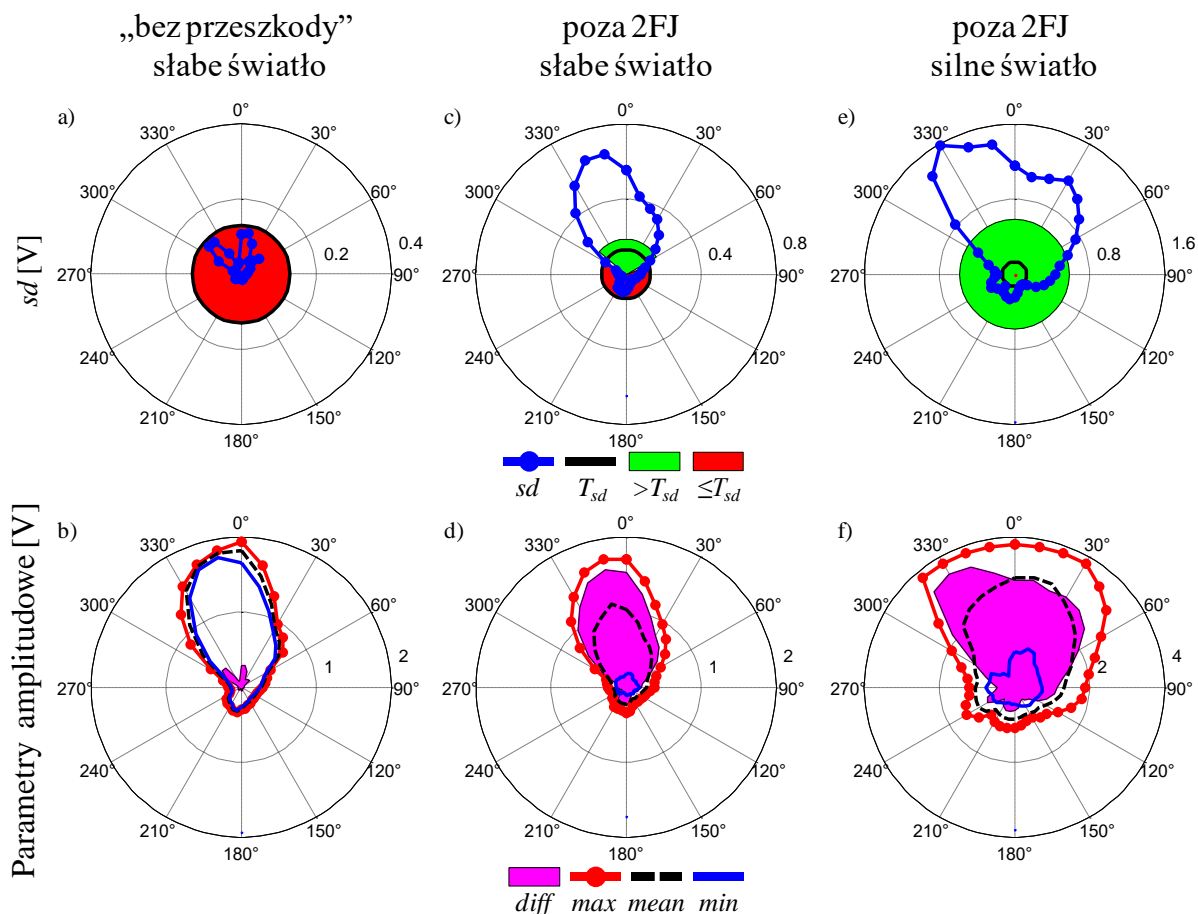
a) *Zachowanie sensora w trakcie obrotów*

Każdy z dwóch rodzajów obrotów z zatkniętą przeszkodą, dla dwóch rodzajów warunków oświetlenia (słabe i mocne światło) przeprowadzono trzykrotnie (w sumie 12 pomiarów). Dodatkowo zmierzono także charakterystykę „bez przeszkody” w polu widzenia czujnika. Wyniki wybranych pomiarów zostały przedstawione na Rys. 4.16. Wybrane spośród trzech pomiarów wykonanych w danej konfiguracji mają środkową jasność zarejestrowaną przez światłomierz.

1) Obrót w kącie φ

Pierwsza część eksperymentu została przeprowadzona w pochmurny dzień, bez przeszkód przed czujnikiem (charakterystyka referencyjna). Średni poziom światła otoczenia podczas pomiaru wynosił 282 ± 53 luksów. Następnie, zamontowano przed czujnikiem przeszkodę imitującą pozę 2FJ i powtórzono obroty. Średnia zmierzona jasność wyniosła 256 ± 21 luksów. Pomiar z przeszkodą wykonano również na drugi dzień, w słonecznych warunkach. Średnia zmierzona jasność wynosiła 1308 ± 144 luksów. Czerwona strefa (jeśli występuje, kolor ciemniejszy) na wykresach polarnych odchylenia standardowego ramki danych $sd(DF)$ (Rys. 4.16a, c, e) przedstawia kąty, dla których sd było poniżej progu sd_{Tp} . Zielona strefa (jeśli występuje, kolor jaśniejszy) pomaga podkreślić pozycje, w których czujnik zauważa przeszkodę ($sd(DF) > sd_{Tp}$).

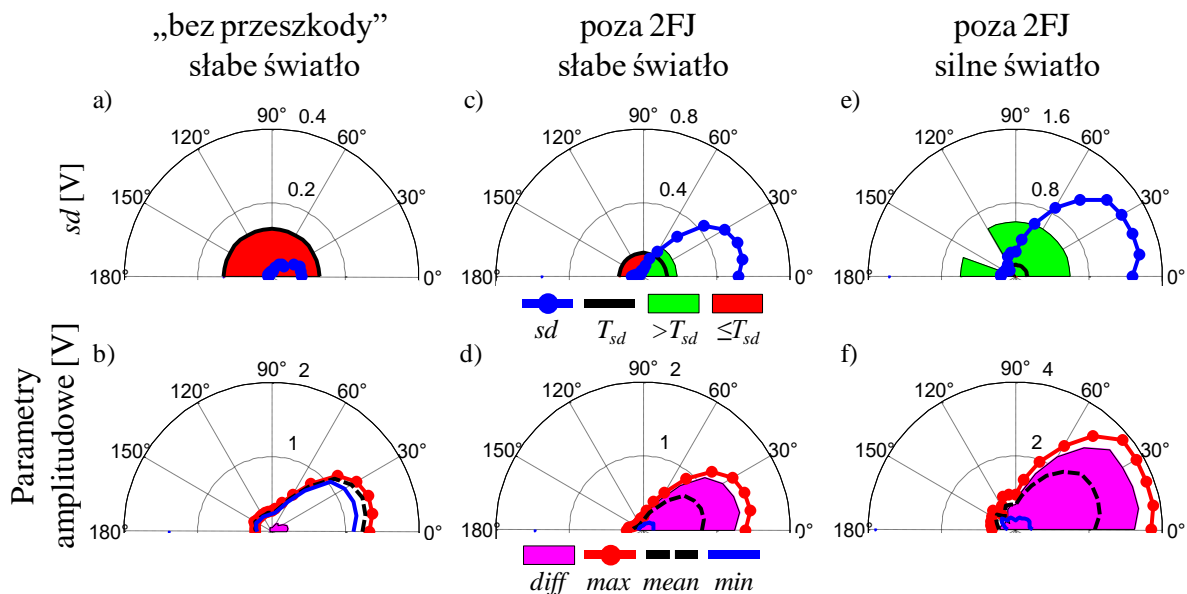
Promień zielonej strefy jest średnią wartością $sd(DF)$ z pomiarów zebranych podczas danego eksperymentu (wszystkie pozycje). Linie kolorów na wykresach związanych z amplitudą (Rys. 4.16b, d, f) przedstawiają maksymalną, średnią i minimalną wartość DF podczas rotacji. Parametr *diff* jest reprezentowany przez zamalowaną strefę w celu podkreślenia jej na wykresie. Zastosowane fotodiody czujnika nasycają się na poziomie około 3,8 V, dlatego wartość parametru *max* może być ograniczona na wykresie.



Rys. 4.16 Wyniki dla obrotu ϕ . Widok z góry, 0° to kierunek okna. Kątowa charakterystyka czujnika w pochmurny dzień bez przeszkód (a, b), z przeszkodą 2FJ w pochmurny dzień (c, d), w słoneczny dzień z przeszkodą 2FJ (e, f). Przebiegi z linią kropkowaną (a, c, e) oznaczają sd , natomiast przebiegi na wykresach (b, d, f) oznaczają parametry związane z amplitudą.

2) Obrót w kącie θ

Pierwsza część tego eksperymentu została przeprowadzona w pochmurny dzień, bez przeszkód przed czujnikiem. Średni poziom światła w otoczeniu podczas pomiaru wyniósł 229 ± 2 luksy. Następnie przed czujnikiem umieszczono pozę imitującą przeszkodę 2FJ. Średnia zmierzona jasność wyniosła 221 ± 16 luksów. W innym dniu, w słonecznych warunkach, przed czujnikiem przymocowano pozę imitującą przeszkodę. Średnia zmierzona jasność wynosiła 837 ± 63 luksów. Wykresy odchylenia standardowego DF przedstawiono na Rys. 4.17a, c, e, natomiast parametry amplitudowe dla różnych warunków świetlnych przedstawiono na Rys. 4.17b, d, f.



Rys. 4.17 Wyniki dla obrotu θ . Widok z boku, 0° to kierunek okna. Kątowa charakterystyka czujnika „bez przeszkód” w pochmurny dzień (a, b) oraz z przeszkodą 2FJ (c, d), w słoneczny dzień z przeszkodą 2FJ (e, f). Przebiegi z linią kropkowaną (a, c, e) oznaczają sd , przebiegi związane z amplitudą są na wykresach (b, d, f).

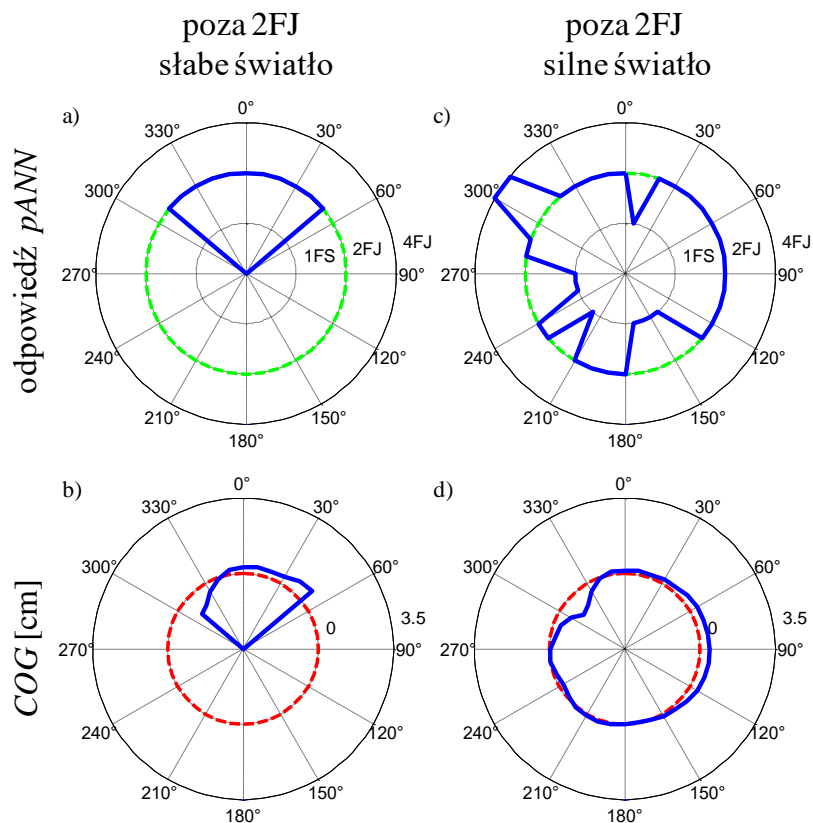
b) Pomiar parametrów związanych z rozpoznawaniem gestów

Dla pomiarów z przeszkodą zamontowaną naprzeciwko czujnika zaprezentowanych w poprzedniej sekcji, dla pozycji, w których $sd(DF) > sd_{Tp}$, wyznaczono także parametry związane ze zdolnością czujnika do rozpoznawania gestów.

1) Obrót w kącie φ

W celu wykrycia statycznej pozy 2FJ w zróżnicowanych warunkach oświetlenia zewnętrznego zastosowano klasyfikator $pANN$. Pogrubiona linia przerywana na wykresach polarnych odpowiedzi $pANN$ (Rys. 4.18a, b) przedstawia oczekiwaną klasę spośród trzech, których klasyfikator został nauczony. Imitacja pozy 2FJ w słabym świetle została rozpoznana w 100% pozycji, w których $sd(DF)$ był większy niż sd_{Tp} . W silnym świetle dokładność rozpoznawania wyniosła 92,67%.

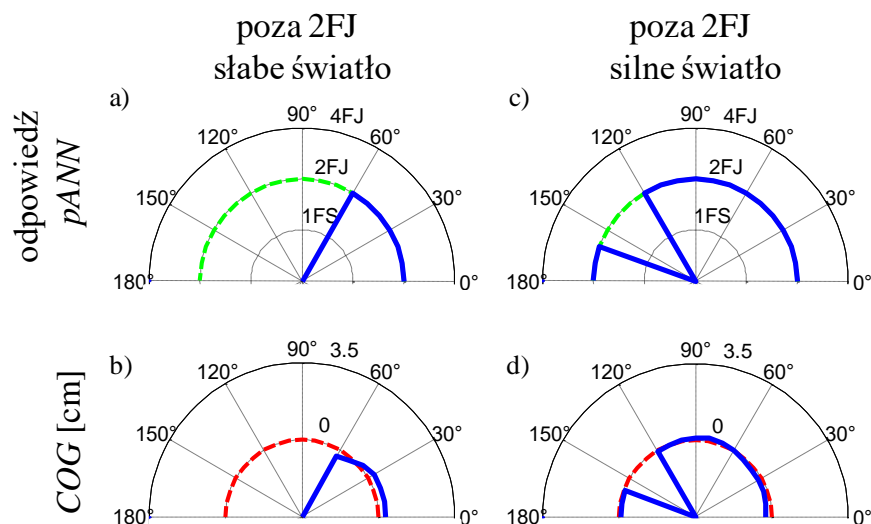
Pogrubiona linia przerywana na wykresach parametru COG wskazuje na oczekiwaną wartość obliczonego położenia pozy w stosunku do czujnika, czyli 0 cm (Rys. 4.18c, d). Aby obserwować bezwzględnie obliczone błędy pozycji w obu kierunkach, poziom 0 cm (w odniesieniu do środka czujnika) nie został typowo umieszczony w środku wykresu polarnego (środek oznacza $COG = -3,5$ cm).



Rys. 4.18 Wyniki dla obrotu φ . Widok z góry, 0° to kierunek okna. Kątowa charakterystyka czujnika w pochmurny dzień z przeszkodą 2FJ (a, b), w słoneczny dzień z przeszkodą 2FJ (c, d). Wykresy (a, c) przedstawiają rozpoznawanie klasy przez $pANN$, gdzie linia przerywana jest oczekiwanym wynikiem; wykresy (b, d) oznaczają obliczone COG , przy czym linia przerywana jest wartością oczekiwaną.

2) Obrót w kącie θ

W przedstawionych przykładowych pomiarach obrotu θ , klasyfikator $pANN$ rozpoznał pozy we wszystkich pozycjach uwzględnionych w analizie (Rys. 4.19a, c). Odpowiadające wartości parametru COG zostały obliczone w tym samym zakresie (Rys. 4.19b, d).



Rys. 4.19 Wyniki dla obrotu θ . Widok z boku, 0° to kierunek okna. Kątowa charakterystyka czujnika w pochmurny dzień z przeszkodą (a, b), w słoneczny dzień z przeszkodą 2FJ (c, d). Wykresy (a, c) przedstawiają rozpoznawanie klas przez $pANN$, gdzie linia przerywana jest oczekiwanym wynikiem; wykresy (b, d) przedstawiają obliczone COG , przy czym linia przerywana jest wartością oczekiwaną.

c) Podsumowanie wyników dla obrotów czujnika

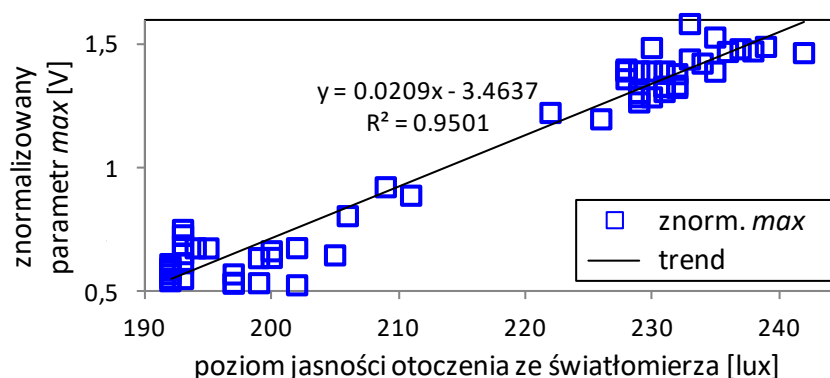
Każdy z pomiarów rotacji czujnika przeprowadzono trzy razy dla każdego rodzaju warunków oświetlenia otoczenia. Dlatego zestawy do analiz (dla słabego i mocnego światła) z obrotu kąтового φ składały się z 108 próbek (3·36) zaś zestawy analiz z obrotu kąтового θ składały się z 57 próbek (3·19). Wyniki średniej jasności zmierzonej podczas eksperymentów, zakresy widzenia przeszkody przez czujnik, dokładność klasyfikatora $pANN$ w pozycjach, w których $sd(DF) > sd_{Tp}$, a także średni błąd wartości obliczonego względem oczekiwanego COG przedstawia Tabela 4.5.

Tabela 4.5 Podsumowanie pomiarów wykonanych w różnych konfiguracjach (warunki świetlne i osie obrotu).

Konfiguracja	jasność [lux]	zakres [°]	zakres [%]	dokł. $pANN$ [%]	COG [cm]
Rotacja φ , słabe światło	278,1±18,2	±56,7±2,9	31,5±2	100±0,0	0,08±0,56
Rotacja θ , słabe światło	215,3±9,5	±31,7±5,8	35,2±3	84,2±24,7	0,13±0,21
Rotacja φ , mocne światło	1263±168	±158,3±25,7	87,9±7	75,8±13	0,02±0,36
Rotacja θ , mocne światło	792,2±59,1	±63,3±17,6	70,3±9	100±0,0	0,07±0,14

4.3.5. Analiza możliwości automatycznego przełączania trybu pracy czujnika optycznego

Jako charakterystykę referencyjną pomieszczenia zastosowano rezultat pomiarów wykonanych z rotacją w kącie θ , bez zamontowanej przeszkody, dla niskiego poziomu światła otoczenia (ok. 200 luksów). Dla każdej z 19 pozycji charakterystyki obliczono maksymalną wartość nieprzetworzonej ramki danych ($max(DF)$). Uzyskany wektor znormalizowano przez podzielenie jego wartości przez średnią z wektora. Następnie bezpośrednio po pomiarze referencyjnym, wykonano trzy powtórzenia pomiaru z rotacją kąta θ oraz tym razem z przeszkodą przyczepioną do uchwyty. Wartości $max(DF)$ z kolejnych ramek danych z zarejestrowanych przebiegów zostały podzielone przez wektor referencyjny (dzielenie element po elemencie). Uzyskane wartości wykreślono następnie na wykresie względem poziomu jasności światła w pomieszczeniu podczas pomiaru, który rejestrowany był przez światłomierz Lutron (Rys. 4.20).



Rys. 4.20 Parametr $max(DF)$ z trzech pomiarów podzielony przez wektor referencyjny wykreślony względem jasności światła otoczenia mierzonego przez światłomierz.

Uzyskane wyniki sugerują, że cień wytwarzany przez przeszkodę 2FJ jest na tyle wąski w stosunku do szerokości czujnika, że nie ma wpływu na relację $max(DF)$ vs mierzona jasność światła otoczenia. Dlatego przyjęto parametr max , jako reprezentujący warunki świetlne panujące w otoczeniu.

Każdy z pomiarów kątowych z przeszkodą został dodatkowo powtórzony dwukrotnie w podobnych warunkach oświetlenia zewnętrznego. W związku z tym, zestaw sześciu pomiarów z obrotami w kątach φ i θ łącznie (każdy składający się z 3 pomiarów wykonanych przy słabym i 3 przy mocnym świetle zewnętrznym) z zamontowaną przeszkodą 2FJ został uwzględniony w analizie kryterium przełączania między trybem aktywnym a pasywnym. Przy każdej pozycji (kącie), w której klasyfikator $pANN$ poprawnie rozpoznawał pozę 2FJ, do wektora cech ramki danych (max , min , $diff$) została przypisany etykieta *bright*

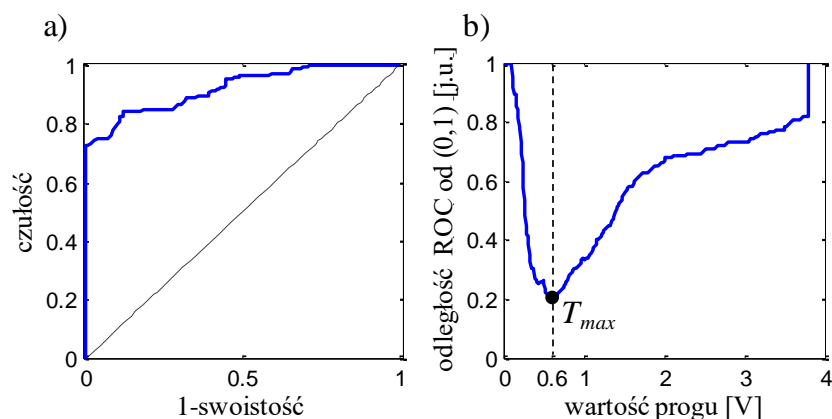
(jasne warunki). W miejscach, w których zauważono nieprawidłową decyzję, przydzielono etykietę *dark* (ciemne warunki). Etykieta *too bright* (zbyt jasno) została przypisana do błędów, które wystąpiły na poziomie nasycenia fotodiod (około 3,8 V).

Metoda drzewa decyzyjnego C4.5 [159] dostępny w oprogramowaniu Weka jako klasyfikator J48 (bez przycinania błędów) zastosowano z ustawieniami domyślnymi w celu oceny warunków, w których czujnik najprawdopodobniej będzie najlepiej działał w trybie pasywnym (klasa *bright*), trybie aktywnym (klasa *dark*) i kiedykolwiek jasność jest zbyt wysoka (klasa *too bright*). Aby zachować warunek przełączania możliwie prostym (optymalizacja szybkości obliczeń), rozważano kryteria jednego parametru. Klasyfikator J48 wybrał parametr *max* z wartością progową, $T_{max}=0,377$ V, która podzieliła zbiór danych wejściowych na klasy *bright* i *dark*. Nie wytworzyła się żadna reguła dla klasy *too bright*, ponieważ w zestawie treningowym znajdowało się więcej poprawnie sklasyfikowanych ramek danych z wartością parametru *max* w obszarze nasycenia niż całkowita liczba przypisanych reprezentacji klasy *too bright*. Otrzymany klasyfikator rozpoznaje próbki klasy *bright* z dokładnością 100%, podczas gdy próbki klasy ciemnej są rozpoznawane z 72,8%. Całkowita dokładność klasyfikatora wynosi 86,36%.

Istnieje jednak asymetria w konsekwencjach dokładności wykrywania klas *dark* (zalecany tryb aktywny) i *bright* (zalecany tryb pasywny). Błąd klasy *dark* powinien być minimalizowany, ponieważ błędna decyzja polegająca na nie włączeniu systemu oświetleniowego (tj. diod LED urządzenia) w warunkach ciemnych może spowodować, że gest w ogóle nie zostanie wykryty (otrzymany profil natężenia światła byłby zbyt płaski). Z drugiej strony, próbki klasy *bright* niepoprawnie sklasyfikowanej jako *dark* doprowadziłyby do zbyt wczesnego włączenia systemu oświetlenia. Kosztem zmniejszenia zakresu wykorzystania pasywnego trybu pracy, pozycje w pewnych sytuacjach mogą być nadal wykrywane w trybie aktywnym. Jednak wartość linii bazowej spowodowanej obecnością światła otoczenia musi być mniejsza niż natężenie światła LED odbitego od dłoni.

Pominąwszy przypadki klasy *too bright*, problem rozpoznania warunków oświetlenia został przeprojektowany na problem z klasyfikacją binarną. Prawdziwe próbki klasy *dark* sklasyfikowane jako *dark* reprezentują próbki prawdziwie pozytywne w klasyfikatorze binarnym, podczas gdy próbki klasy *bright* sklasyfikowane jako *bright* są próbkami prawdziwie negatywnymi. Dlatego na krzywej ROC zaobserwować można wpływ

dostosowania progu T_{max} w zakresie od 0 do 3,8 V na czułość i swoistość klasyfikatora binarnego (Rys. 4.21a). Skok zmiany wartości T_{max} podczas dostosowywania wynosił 0,01 V.



Rys. 4.21 a) Krzywa ROC klasyfikatora warunków oświetlenia. b) Wykres optymalnego punktu krzywej ROC.

Obliczono także odległość między każdym punktem krzywej ROC a doskonałym klasyfikatorem binarnym (0,1) (Rys. 4.21b). Najbliższy punkt, który wskazuje optimum, został osiągnięty dla $T_{max}=0,6$ V. Wartość ta odpowiada 142,7 luksom mierzonym przez sondę światłomierza skierowaną do źródła światła pod tym samym kątem, co fotodiody i obliczona została na podstawie wzoru funkcji liniowej uzyskanej z analizy regresji liniowej parametru *max* względem poziomu jasności wskazywanego przez światłomierz.

Po zastosowaniu nowego, dopasowanego progu utworzono tablicę pomyłek klasyfikatora (Tabela 4.6). Obliczona czułość klasyfikatora wyniosła 84,11%, podczas gdy swoistość 88%. Przed korektą progu decyzyjnego wartości te wynosiły odpowiednio 69,54% i 100%. Ogólna dokładność klasyfikatora warunków oświetlenia wynosi 85,15%.

Tabela 4.6 Tablica pomyłek dla warunków operacyjnych po dopasowaniu progu klasyfikacji.

	Klasyfikowane jako <i>dark</i>	Klasyfikowane jako <i>bright</i>	
Prawdziwie <i>dark</i>	(TP) 127	(FN) 24	Czułość = 127/151 = 84,11%
Prawdziwie <i>bright</i>	(FP) 21	(TN) 154	Swoistość = 154/175 = 88%

4.3.6. Dyskusja i podsumowanie

Wykorzystanie klasyfikatora *aANN* do rozpoznawania przypadków podzbioru testowego z pasywnego zbioru danych nie pozwoliło na uzyskanie wysokiej dokładności klasyfikacji zwartych pól dłoni (52,82%). W związku z tym zdecydowano o projekcie i treningu dedykowanego dla trybu pasywnego klasyfikatora, *pANN*, którego dokładność

klasyfikacji podzbioru testowego pasywnego zbioru danych wyniosła 98,76%. Zestawienie to uwidacznia, iż w zależności od warunków oświetlenia otoczenia, należy wybrać nie tylko tryb pracy czujnika, ale także zastosować odpowiedni model klasyfikacyjny (klasyfikator ANN), aby zmaksymalizować wydajność czujnika rozumianą jako efektywność energetyczną i wysoką dokładność klasyfikacji póź.

Przeprowadzono analizę możliwej wartości progowej dla rozróżnienia stanów gest / brak gestów postrzeganych przez czujnik optyczny. Próg sd_{Tp} równy 0,13 V został eksperymentalnie wybrany, ponieważ prawie 95% ramek danych z czterech pomiarów kątowych (każdy reprezentujący klasę „bez przeszkody”) charakteryzowało się mniejszą wartością odchylenia standardowego. Oznacza to jednak, że w niektórych rzadkich przypadkach czujnik może rozpocząć analizę i klasyfikację „pustych” ramek danych (tj. gdy naprzeciwko czujnika nie znajduje się żadna przeszkoda).

Właściwości czujnika w różnych warunkach oświetlenia zostały zmierzone poprzez wprowadzenie kontrolowanej zmiany orientacji czujnika w odniesieniu do źródła światła. Zmierzone profile światła dla każdej orientacji, po zastosowaniu wartości progowej, sd_{Tp} , oznaczono jako otrzymane w odpowiednich (poprawna klasyfikacja profilu natężenia światła jako przeszkody 2FJ) lub w nieodpowiednich (błędna klasyfikacja) warunkach oświetlenia dla czujnika. Podczas eksperymentów czujnik był umieszczony centralnie w odległości 2,6 m od okna o szerokości 3,1 m, stąd oczekiwano bezpośredniego światła zewnętrznego w zakresie około $\pm 30^\circ$. Wyniki eksperymentów pokazują średni zakres, w którym (na podstawie trzech pomiarów) wartość $sd(DF)$ była większa niż sd_{Tp} . W przypadku słabego światła otoczenia czujnik zaobserwował (ale niekoniecznie poprawnie sklasyfikował) każdą pozę w zakresie kątów w obrębie φ około $\pm 57^\circ$ (278 ± 18 luksów) i dla zakresu kątów w obrębie θ około $\pm 32^\circ$ (215 ± 10 luksów). Dla silniejszego światła wartości te wzrosły odpowiednio do $\pm 158^\circ$ (1263 ± 168 luks) i $\pm 63^\circ$ (792 ± 59 luksów). Podane zakresy odnoszą się do kierunku źródła światła (słońca). Gdyby czujnik był używany w multimedialnych okularach lub goglach VR, wówczas obrót w kącie φ byłby bardziej naturalny i częstszy niż pod kątem θ . W uproszczonym oszacowaniu, dla uśrednionych warunków nasłonecznienia w ciągu dnia, można określić średni zakres widoczności na $\pm 107,5^\circ$ (średnia z zakresów widoczności uzyskanych dla obrotu φ dla słabego i mocnego natężenia światła otoczenia). Dla użytkownika, który czasowo równomiernie pod wszystkimi kątami z zakresu pełnego obrotu układu swoją głowę (a więc i czujnik) w stosunku do kierunku słońca, może to dać

przybliżoną ocenę oszczędzania energii (przełączanie czujnika z trybu aktywnego na tryb pasywny) przez 60% czasu w ciągu dnia.

Przy obliczaniu średniej dokładności rozpoznawania póż do podsumowania, które prezentuje Tabela 4.5, uwzględnione były decyzje klasyfikatora $pANN$ tylko w pozycjach, dla których przekroczony został próg sd_{Tp} . Pomiar dla danego kąta i rodzaju światła otoczenia (ciemne / jasne) wykonano po trzy razy, a wyniki uśredniono. W słabych warunkach oświetleniowych dokładność klasyfikacji pozy 2FJ wynosiła 84,2% dla obrotów θ i 100% dla obrotów φ . W silniejszym świetle dokładności były odpowiednio na poziomie 100% i 75,8% dla obrotów w kątach θ i φ . Średnia z tych wartości (87,9%) różni się znacznie od wysokiej dokładności rozpoznawania osiągniętej w podzbiórce testowym z pasywnego zbioru danych (98,76%). Przyczyną tego może być fakt, iż eksperymenty użyteczności metody rozpoznawania póż dłoni zostały przeprowadzone dla zróżnicowanych orientacji czujnika optycznego względem kierunku źródła światła słonecznego (obroty), podczas gdy podzbiór testowy został uzyskany w sprzyjających warunkach (światło słoneczne prostopadłe do czujnika).

Istnieje wiele prac opisujących technologie, która może być wykorzystana do pasywnego wykrywania gestów, ale często nie zawierają one badań dotyczących dokładności ich wykrywania [96][105][160]. Badania zawierające element ewaluacji przedstawiają natomiast zróżnicowane wskaźniki klasyfikacji gestów. Matryca 2x2 czujników PIR rozpoznaje 4 trajektorie ruchu z dokładnością 77% [75]. Czujnik AllSee wykazuje dokładność klasyfikacji na poziomie 97% dla zestawu 8 gestów przy wykorzystaniu sygnałów RFID [98]. Optyczny czujnik gestów oparty na matrycy fotodiod (3x3) opisany w [73] rozpoznaje 10 gestów z dokładnością 98%. Klasyfikator tego czujnika został przeszkolony i oceniony pod kątem zróżnicowanych warunków oświetlenia zewnętrznego, które podzielono na dwie ogólne kategorie, mianowicie światło i ciemność. Praca ta nie zawiera jednak analizy wpływu różnych kątów oświetlenia czujnika na dokładność rozpoznawania gestów.

Liczby przedstawiane dotychczas dla czujnika optycznego o liniowej budowie dotyczą wykrywania pozy ręki. Czujnik nadal wymaga klasyfikatora wyższego poziomu, który pozwoliłby na zbudowanie zestawu gestów w oparciu o swoją zdolność do rozpoznawania póż dłoni oraz jej szacowaną lokalizację w kolejnych cyklach próbkowania. Dokładne wykrywanie położenia palców dłoni w odniesieniu do lokalizacji czujnika dla różnych warunków oświetlenia może być ważne dla rozpoznawania gestów

dynamicznych. W związku z tym, w trakcie eksperymentów związanych z obrotami badano także dokładność określenia pozycji wzdłużnej sztucznej pozy 2FJ wyznaczonej przez wartość parametru *COG*, względem rzeczywistego ułożenia przeszkody nad czujnikiem $x_o=0$ cm (poza powyżej środka czujnika). Średnia wartość parametru *COG* dla obrotów w słabych i mocnych światłach wynosiła odpowiednio $0,08\pm 0,56$ cm i $0,02\pm 0,36$ cm. Średnie wartości obliczonej pozycji przeszkody podczas obrotów w kącie θ w warunkach słabego i silnego oświetlenia zewnętrznego wynosiły odpowiednio $0,13\pm 0,21$ cm i $0,07\pm 0,14$ cm. Znacznie niższe odchylenie standardowe pozycji w obrocie w θ można przypisać temu, że kąt między wzdłużną częścią przeszkody a kierunkiem źródła światła w tym eksperymencie nie zmieniał się. Obserwowalne przesunięcia do dodatnich i ujemnych wartości *COG* w obrocie φ są spowodowane przesunięciem cienia przeszkody. Pokazuje to, że obliczona pozycja palców w trybie pasywnym silnie zależy od wzajemnego usytuowania czujnika i źródła światła. Jednak cecha ciągłej lokalizacji ręki w stosunku do czujnika, jak w [96], czyni badane urządzenie atrakcyjnym rozwiązaniem wśród innych prostych czujników gestów. Jest to znaczna przewaga nad niektórymi czujnikami, które obsługują jedynie gesty dyskretne [54][73][74]. Dokładne rozpoznawanie pozycji ręki i jej umiejscowienie w stosunku do czujnika można wykorzystać do zdefiniowania kilku dynamicznych gestów, takich jak: palec poruszający się od lewej do prawej; dwa palce trzęsące się przed czujnikiem i wiele innych. Będą one przedmiotem badań opisanych w następnej sekcji niniejszego rozdziału.

W pasywnym trybie pracy czujnik okazał się niezdolny do prawidłowej pracy nie tylko w ciemności (co jest oczywiste), ale także w silnych warunkach oświetleniowych (silne cienie, nasycenie fotodiod). Aby znaleźć optymalny warunek przełączania pomiędzy pasywnym i aktywnym trybem pracy czujnika, zastosowano klasyfikator drzewa decyzyjnego J48 z oprogramowania Weka. Warunki oświetlenia otoczenia zostały przedstawione jako wektor cech *max*, *min* oraz *diff* i zostały przypisane do jednej z trzech klas: *dark*, *bright* i *too bright*. Klasyfikator o głębokości jednego węzła wskazał na parametr *max* o optymalnej wartości progowej $T_{max}=0,377$ V i pozwolił na klasyfikację warunków świetlnych z dokładnością 86,36%. Jednak ze względu na asymetrię konsekwencji błędnej klasyfikacji klas ciemnych i jasnych wykorzystano analizę czułości i swoistości. Dodatkowo przypadki klasy *too bright* zostały odrzucone z badania, ponieważ w trakcie badań zgromadzono więcej poprawnie sklasyfikowanych próbek z parametrem *max* w regionie

nasylenia niż wynosi ogólna liczba przypisanych próbek klasy *too bright*. Dlatego problem został rozwiązany ponownie, tym razem dla klasyfikatora binarnego. Za pomocą drzewa decyzyjnego nowa wartość progowa T_{max} została wyznaczona jako 0,6 V, a czułość i swoistość nowego klasyfikatora wyniosły odpowiednio 84,11% i 88%. W sytuacjach, gdy parametr $max(DF)$ osiąga obszar nasylenia, czujnik może wywołać alarm, aby wskazać, że wyniki rozpoznawania pozy mogą być niedokładne. Nowa wartość progowa zapewnia równowagę pomiędzy minimalizacją liczby póż przedstawionych w warunkach rzeczywistych ciemnych, sklasyfikowanych jako jasne (fałszywie ujemne) i prawdziwe jasne, sklasyfikowane jako ciemne (fałszywie pozytywne). Pozy prezentowane czujnikowi w warunkach z kategorii fałszywie negatywne najprawdopodobniej zostaną niezauważone, ponieważ wzór światła zarejestrowany przez fotodiody byłby bardzo płaski (niski poziom światła otoczenia i słaby cień). Pozy przedstawione w warunkach z kategorii fałszywie pozytywnych mogą zostać wykryte, gdyby wartość linii bazowej spowodowana światłem otoczenia była mniejsza niż rejestrowane natężenie światła LED odbijanego od dłoni.

Ważnym czynnikiem opisującym właściwości urządzeń elektrycznych jest poziom zużycia energii elektrycznej. Typowy pobór mocy przez aktywne czujniki gestów o prostej budowie wynosi znacznie poniżej 1 W jak w urządzeniu Okuli, które można zredukować do około 100 mW [67] zaś zużycie prądu w efektywnych konstrukcjach może nie przekraczać 10 mA jak np. 3,78 mA w czujniku o częściowo otwartej komorze [54]. Większość pasywnych czujników gestów o prostej budowie często wymaga jednak kilku rzędów wielkości mniejszego prądu do działania. Czujnik 2x2 PIR wymaga do działania poniżej 50 μ W [75]. Pobór mocy czujnika AllSee wykorzystującego sygnały bezprzewodowe zmierzono dla dwóch typów prototypów. Prototyp oparty na ADC wykorzystuje 28,91 μ W, zaś ten na podstawie układu analogowego jedynie 5,85 μ W do wykrywania 15 gestów na minutę [98]. Średni pobór prądu pojedynczej fotodiody rozważanego w niniejszej rozprawie czujnika optycznego o liniowej budowie wynosi 1,1 mA. Biorąc pod uwagę $f_s=50\text{Hz}$ i $D=1,5\%$, całkowity pobór prądu przez 8 fotodiod o kluczowanym zasilaniu wynosi 165 μ A (825 μ W). W aktywnym trybie pracy czujnika o budowie liniowej zużycie prądu wynosi średnio 2,48 mA (12,4 mW). Dlatego przejście z aktywnego do pasywnego trybu pracy prowadzi do zmniejszenia mocy wykorzystywanej przez elementy optoelektroniczne czujnika liniowego o 93,34% (wartość ta pozostaje stała wraz ze zmianą częstotliwości próbkowania czujnika).

Ponieważ przełączanie pomiędzy aktywnym i pasywnym trybem pracy czujnika zależy tylko od wartości $\max(DF)$, decyzja dotycząca wykorzystania diod LED nie wymaga wielu obliczeń. Dla mikrokontrolera wykorzystywanego w czujniku gestów o liniowej budowie, zmierzono czas wymagany do podjęcia decyzji (warunek przełączania) i na przeprowadzenie dodatkowego próbkowania. Czas pomiędzy ukończeniem próbkowania w pasywnym i aktywnym trybie pracy (oddzielony przez obliczenie warunku przełączania) wynosił około $500\mu s$. Dlatego próbkowanie pasywne może być wykonywane za każdym razem jako tanie energetycznie i referencyjne, zaś próbkowanie aktywne dodatkowo w tym samym cyklu zegara ale tylko po spełnieniu warunku T_{max} . Wybór optymalnego trybu pracy podczas wykonywania gestu nie powinien wpływać na działanie czujnika, ponieważ kiedy działa on z częstotliwością rzędu 50Hz, okres (20ms) trwa znacznie dłużej niż $500\mu s$.

Przeprowadzone badania świadczą o tym, że pasywny tryb działania optycznych czujników gestów może być rozważany w pewnym zakresie warunków oświetlenia otoczenia, zamiast aktywnego trybu pracy o większym poborze mocy. Bardzo znaczna redukcja zapotrzebowania na prąd czujnika w trybie pasywnym umożliwia urządzeniom mobilnym wykorzystującym taki czujnik gestów, np. multimedialne okulary czy gogle VR na dłuższy czas operacyjny, bez znaczącej redukcji wydajności czujnika.

4.4. Rozpoznawanie gestów z wykorzystaniem rekurencyjnych sieci neuronowych

Rozpoznawanie gestów, w odróżnieniu od klasyfikacji pól, wiąże się z analizą sekwencji danych gromadzonych w czasie. Weryfikację wybranej metody rozpoznawania gestów rejestrowanych przez czujnik o liniowej budowie przeprowadzono na przykładach z nowej bazy danych, zawierającej sekwencje. Na potrzeby jej opracowania odwołano się do odpowiednich kategoryzacji, utworzono protokoły pomiarowe i zgodnie z nimi przeprowadzono eksperymenty z udziałem ochotników. Następnie w podrozdziale opisana została procedura uczenia modelu na podstawie danych ze zgromadzonej bazy gestów oraz opisane zostały rezultaty dokładności wybranej metody dla pomiarów z czujnika optycznego.

4.4.1. Baza gestów dłoni

Gesty uwzględnione w niniejszych badaniach są typu dyskretnego, co oznacza, że reakcja systemu będzie następować dopiero po zakończeniu pokazywania gestu przez użytkownika. Zgodnie z taksonomią Wobbrocka jest to kategoria wymiaru *flow* [30].

Rozważane gesty zostały podzielone na trzy grupy ze względu na adekwatne kategorie wymiaru *form* Wobbrocka. Poniższa charakterystyka każdej z grup zawiera także opisy konkretnych gestów wykonywanych podczas eksperymentów. Wszystkie nazwy (poza jedną: Cut [65]) opisujące rozważane gesty zostały zdefiniowane przez autora rozprawy.

a) Poza statyczna

Przypadki z tej grupy stanowią gesty gdzie konkretna poza jest umieszczona naprzeciwko sensora na krótki czas (około sekundę) i zabrana z powrotem z jego pola widzenia. Zasadą jest, iż zarówno poza jak i pozycja ręki (nie licząc fazy wkładania i wyjmowania dłoni z pola widzenia sensora) pozostają niezmiennie w trakcie trwania gestu. W celu zademonstrowania, jak liczba gestów z tej grupy może zostać rozszerzona, część gestów była pokazywana na środku osi symetrii sensora ($x_o=0$ cm), na wysokości $h=3$ cm; część również na środku ale dalej od sensora ($h=5$ cm) oraz część nad krawędziami sensora ($h=3$ cm, $x_o=\pm 3$ cm). Gesty, dla których miejsce wykonania względem sensora ma znaczenie (czyli rozróżnia gesty między sobą) są klasyfikowane pod kategorią *world-dependend* z wymiaru *binding* Wobbrocka.

b) Poza statyczna i ścieżka

Dla gestów z drugiej kategorii, poza ręki wciąż pozostaje statyczna, lecz zmienia się ścieżka / trajektoria zakreślana przez dłoń przed czujnikiem. W przeprowadzonych eksperymentach na gesty te składają się przesunięcia dłoni (ang. *swipes*) wzdłuż osi x (rozdzielany jest także kierunek w lewo / prawo oraz prędkość wolna / szybka) wykonywane na wysokości $h=3$ cm; gesty związane ze zbliżaniem i oddalaniem dłoni (ang. *zoom in*, *zoom out*) z odległości $h=1$ cm nad sensorem aż do $h>10$ cm (poza zasięgiem pola widzenia czujnika); potrząsanie dłonią, gdzie poza dłoni oscyluje gwałtownie 3 razy w osi horyzontalnej oraz wertykalnej (osobny gest) pozostając cały czas w polu widzenia czujnika; zakreślanie trajektorii elipsy utrzymując stałą pozę dłoni.

c) Poza dynamiczna

Gesty z trzeciej grupy charakteryzują się tym, iż dłoń zmienia pozę w trakcie ich wykonywania lecz pozostaje w tym samym miejscu. Cut naśladuje cięcie nożyczkami i określa sekwencję póż: 2FS, 2FJ oraz 2FS wykonywaną podczas tego samego pojawienia się dłoni w polu widzenia czujnika. Gest Dbcut (double cut - podwójne cięcie) jest to gest Cut wzbogacony o dodatkowe cięcie czyli o dołożenie póż 2FJ i 2FS do sekwencji Cut.

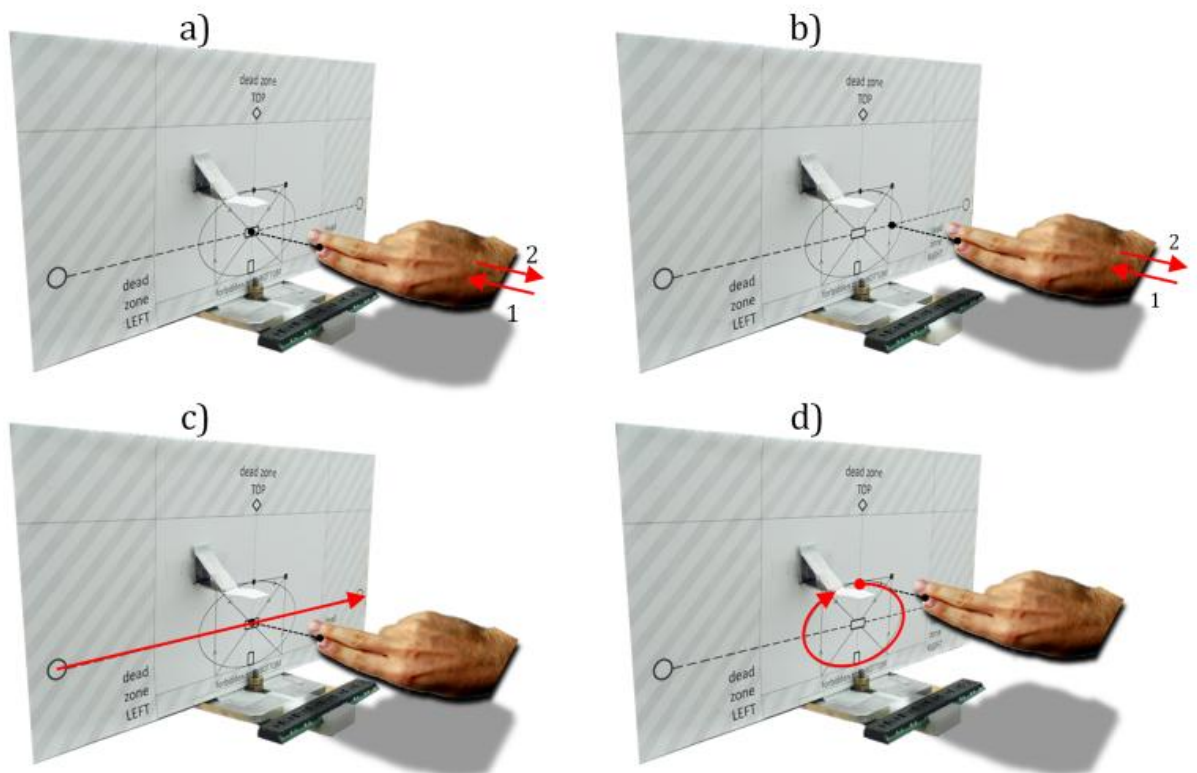
W kontekście interakcji człowieka z maszyną można te gesty interpretować jako równoważniki pojedynczego i podwójnego kliknięcia przycisku myszy komputerowej. Ostatni z gestów tej kategorii, Chopin, polega na umieszczeniu całej dłoni nad sensorem i asynchronicznym ruchu wszystkimi palcami imitującym grę na wirtualnym pianinie.

Biorąc pod uwagę ilość stopni swobody gestów kategorii *poza statyczna* oraz *poza statyczna i ścieżka* całkowita liczba analizowanych gestów wynosi 27. Tabela 4.7 podsumowuje zbiór zaproponowanych gestów.

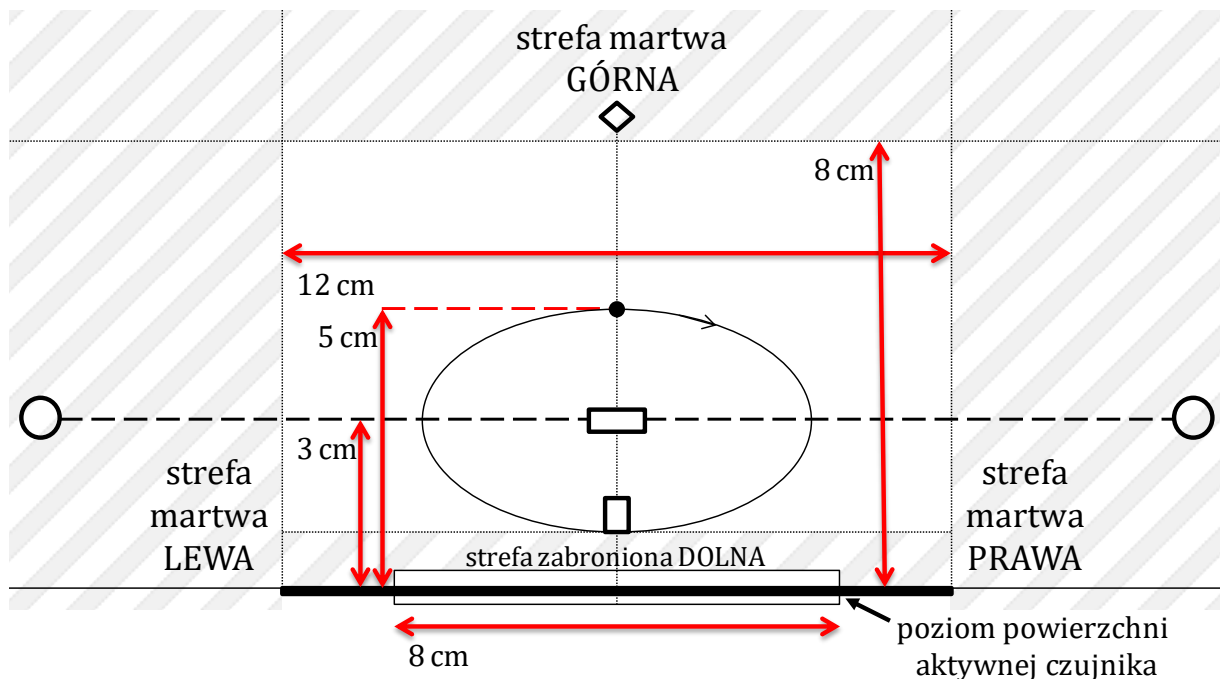
Tabela 4.7 Podsumowanie zbioru gestów rozważonych w badaniach.

Zdefiniowana nazwa gestu	Pozy biorące udział	Lokalizacja	Ścieżka	Kategoria wymiaru form Wobbrocka
1FS	1FS	Środek		Poza statyczna
2FS	2FS	Środek		
2FJ	2FJ	Środek		
2FJside	2FJ	Prawo / Lewo		
2FJhigh	2FJ	Środek wysoko		
4FJ	4FJ	Środek		
1FSslow	1FS		Prawo / Lewo	Poza statyczna i ścieżka
2FSslow	2FS		Prawo / Lewo	
2FJslow	2FJ		Prawo / Lewo	
2FJfast	2FJ		Prawo / Lewo	
4FJfast	4FJ		Prawo / Lewo	
2FJzoom	2FJ	Zbliżanie/Oddalanie		
4FJzoom	4FJ	Zbliżanie/Oddalanie		
2FJshakeVer	2FJ	Wertykalne potrząsanie		
2FJshakeHor	2FJ	Horyzontalne potrząsanie		
2FJellipse	2FJ		Elipsa	
Cut	2FS, 2FJ	Środek		Poza dynamiczna
Dbcut	2FS, 2FJ	Środek		
Chopin	4FJ falując palcami	Środek		

Pomiary wykonywane były w trakcie pogodnego dnia, w warunkach naturalnych - nie nakładano na sensor żadnych dodatkowych przesłon (poza zawsze obecnym blokiem optycznym) zasłaniających potencjalnie zmienne światło otoczenia. Czujnik pracował w trybie różnicowym, DM (AM-PM), z częstotliwością próbkowania 100Hz (celem dokładnej rejestracji trajektorii gestu). Na potrzeby eksperymentu czujnik gestów został zamontowany w specjalnym uchwycie (fotodiodami do góry) na biurku, naprzeciwko stanowiska dla ochotników umieszczanych na krześle, aby mogli oni wygodnie kontrolować swoje ruchy (Rys. 4.22). Arkusz tektury z wydrukowanymi punktami odniesienia i trajektoriami dla poszczególnych gestów został zamontowany prostopadle do powierzchni czujnika aby zwiększyć powtarzalność gestów wykonywanych przez ochotników (Rys. 4.23).



Rys. 4.22 Przykładowe gesty. Poza statyczna: 2FJ, 2FJsideR (a, b); poza statyczna i ścieżka: 2FJslowR, 2FJellipse (c, d). Czerwone strzałki wskazują zadaną trajektorię ręki, linia przerywana wskazuje miejsce na tekturze, na które wskazuje dłoń. Czarny element na dole każdego obrazu to czujnik gestów z nakładką. Istotne jest położenie cienia wytwarzanego przez rękę, ponieważ wskazuje on górną projekcję a więc lokalizację dłoni wzdłuż osi x (ważne dla gestów poza statycznej). Rozmiar dłoni na rysunku może nie odzwierciedlać dokładnej skali w stosunku do czujnika.



Rys. 4.23 Obraz z trajektoriami i pozycjami do wykonywania gestów, który przyklejony został do tektury i zamontowany na uchwycie z czujnikiem optycznym. Powierzchnia aktywna fotodiody czujnika ($y=0$) została umieszczona na poziomie czarnej pogrubionej linii (u dołu obrazu). Czerwone strzałki naniesiono w celach referencyjnych (nie były drukowane).

26 dorosłych ochotników (13 kobiet, 13 mężczyzn, wiek: $34,3 \pm 10,4$ lat) wzięło udział w eksperymencie polegającym na powtórzeniu każdego ze zdefiniowanych gestów 20 razy. Dzięki temu, za pomocą czujnika liniowego zarejestrowanych zostało ponad 500 reprezentacji każdego gestu. Łącznie uzyskano zatem ponad 13000 sekwencji, które były na bieżąco przesyłane (ramka po ramce) i zapisywane w pamięci komputera PC w celu dalszego przetwarzania.

Niektóre z metod rozpoznawania gestów opisanych w rozdziale 2 oczekują podania jako wejścia do modelu pozy dłoni w sposób niejawny (np. przez lokalizacje poszczególnych stawów dłoni). W badaniach przeprowadzonych w niniejszej rozprawie porównana zostanie wydajność modelu rekurencyjnych sieci neuronowych, RNN, z wejściami w postaci niejawnych (nieprzetworzone dane lub wektory cech) i jawnych (kody rozpoznanych póz i lokalizacja dłoni) informacji na temat aktualnej pozy dłoni. Podczas analizy dokładności detekcji gestów przez model bazujący na RNN rozważano zatem trzy rodzaje reprezentacji danych. Wynikowe zbiory składają się z dokładnie tych samych sekwencji wykonanych przez tych samych ochotników, lecz różnią się stopniem przetworzenia ramki danych. Sekwencje zostały wyodrębnione i przetworzone za pomocą oprogramowania MATLAB®.

a) Gest jako nieprzetworzone dane

Ponieważ zdefiniowano jako sekwencję stempli czasowych opisujących pozy (rozdział c)), najbardziej podstawową reprezentacją gestu zarejestrowanego przez czujnik o budowie liniowej jest tablica 2D. Ta macierz złożona z ramek danych zawiera wartości profili natężenia światła z kolejnych cykli próbkowania (8 kolumn, każda z wartością natężenia światła z jednej fotodiody czujnika optycznego, rzędów tyle ile wykonano cykli próbkowania w trakcie obecności dłoni w polu widzenia czujnika). Dane w tej postaci nie zawierają żadnej interpretacji prezentowanej pozy ani pozycji ręki, dlatego też model rozpoznawania gestów ma za zadanie samemu się ich nauczyć. Taki zestaw danych będzie określany jako nieprzetworzone dane (ang. *raw data*).

b) Gest jako wektory cech

Gesty można również traktować jako serię elementarnych póz w czasie i przestrzeni. Dlatego, dla każdego ze stempli czasowych we wszystkich sekwencjach obliczono zestaw cech. Są to cechy, na podstawie których zaproponowano system rozpoznawania gestów dla klasyfikacji pozy w publikacji autora rozprawy [156]: liczba pików w DF (*PKS*)

oraz wybrane wejścia do dwuwarstwowej sieci neuronowej (Sekcja 4.2)(środek ciężkości DF (*COG*); *mean*(DF); *sd*(DF) i liczba wartości w DF mniejsza niż $2 \cdot sd(DF)$). Aby umożliwić modelowi śledzenie ruchów ręki z większą precyzją, opisany zestaw funkcji został uzupełniony o *max*(DF). Stempel czasowy drugiego zestawu danych składa się zatem z 6 kolumn a ten rodzaj danych będzie określany jako wektory cech (*features*). Dane te zawierają jawną informację o pozycji ręki (*COG*, *max*) w stosunku do czujnika i wciąż niejawną ale przetworzoną do postaci cech informację dotyczącą pozy.

c) Gest jako wektory cech wysokiego poziomu

Trzeci zestaw danych składa się jedynie z 3 cech. Pierwszą cechą jest rozpoznana poza (1FS, 2FS, 2FJ lub 4FJ) uzyskana z GRS z dwuwarstwową siecią neuronową [156]. Pozostałe dwie cechy to *COG* i *max* odzwierciedlające pozycję ręki względem czujnika. Ten zbiór danych będzie określany jako cechy wysokiego poziomu / *HLfeatures* (ang. *high level features*).

Symetria konstrukcyjna czujnika pozwala na wykonanie operacji rozszerzenia zbioru danych (ang. *data augmentation*). Ustalono, iż transformacje stosowalne do tego typu danych i gestów to: odbicie lustrzane (względem osi *y*), odwrócenie czasu (odtworzenie sekwencji od tyłu) i odbicie lustrzane i czasowe jednocześnie. Należy jednak dane transformacje ostrożnie stosować, ponieważ w przypadku gestów przesunięcia (*swipe*) lub powiększenia (*zoom*) zostaną one przekształcone w instancję klasy przeciwnej (np. przesunięcie 1FSsłow w prawo zamiast przesunięcia 1FSsłow w lewo). W takim wypadku należy przypisać gestom po transformacji odpowiednią etykietę (np. klasy przeciwnej). Dodatkowo zauważono, iż gest elipsy może podlegać tylko jednej z podanych transformacji (jednoczesne odbicie lustrzane i czasowe). W związku z tym zdecydowano o przeprowadzeniu dla każdego gestu wszystkich możliwych dla niego transformacji. Dzięki temu powstały dodatkowe instancje gestów i, celem zachowania równomiernej liczby przypadków w klasach, wybrano z nich i na podstawie oryginalnych danych po 1000 sekwencji na każdy gest. Średnia długość sekwencji wynosiła $58,66 \pm 40,91$ stempli czasowych (minimum = 4, maksimum = 332).

Podczas eksperymentów, gdy uczestnicy byli proszeni o wykonywanie powolnych i szybkich gestów, nie zdefiniowano zadanego tempa ruchu. Dlatego wykonywali oni gesty zgodnie z indywidualnym poczuciem szybkości. W rezultacie, pod względem szybkości



ruchu dłoni, wolne gesty wykonywane przez niektórych uczestników były porównywalne do szybkich gestów dla innych uczestników. Dlatego podczas analizy wprowadzono sztywny próg definiujący dopuszczalny czas trwania wolnego i szybkiego przesunięcia aby uczony model dostał przykłady różnych klas o zbiorach rozłącznych pod względem szybkości ruchu. Proóg wyznaczony był na podstawie liczby stempli czasowych, dla której klasyfikator binarny wykazał maksymalną czułość. Uzyskana wartość progu to $t_{TS}=21$ stempli czasowych. Przesunięcia zarejestrowane jako szybkie ale liczące więcej niż t_{TS} stempli czasowych zostały odrzucone. Analogiczna procedura zastosowana została do odfiltrowania przesunięć wolnych. Ilość transformacji przy rozszerzaniu danych pozwoliła na uzyskanie znacznie więcej niż 1000 przykładów dla większości gestów dlatego zastosowanie progów przy gestach związanych z przesunięciami nie wpłynęło na końcową liczbę instancji tych klas. Ponadto, należy zauważyć, iż oszacowanie szybkości ruchu dłonią może być realizowane analitycznie, poza modelem RNN, tak aby użytkownik mógł ustawić próg w zależności od swoich upodobań. Jednak w przeprowadzonych analizach zbadano również, w jaki sposób modele RNN są w stanie rozróżnić te same gesty wykonywane w różnych skalach czasowych.

Utworzone zbiory danych podzielono klasycznie na zbiory treningowe, walidacyjne i testowe o stosunku liczności 70%/15%/15% z wykorzystaniem losowania warstwowego (ang. *stratified sampling*) co pozwoliło na uzyskanie zbiorów doskonale zbalansowanych co do liczności przykładów w klasach. W odpowiednich zbiorach (np. treningowym) danego typu reprezentacji danych znajdowały się dokładnie te same instancje zarejestrowanych gestów aby wyniki uczenia modeli były porównywalne. Całkowita liczba gestów w zbiorach treningowych, walidacyjnych i testowych wyniosła odpowiednio 18900 (po 700 obserwacji na klasę) i po 4050 (po 150 obserwacji na klasę).

4.4.2. Uczenie modelu RNN

W odróżnieniu od problemu klasyfikacji pozy, w zagadnieniu detekcji gestów ważna jest także informacja o położeniu lub trajektorii ręki w kolejnych cyklach próbkowania. Dlatego jako metodę rozpoznawania gestów wybrano model do nadzorowanego uczenia sekwencji w postaci rekurencyjnej sieci neuronowej. Fazy uczenia i ewaluacji modelu zostały przeprowadzone w trybie sekwencja do wektora (ang. *sequence to vector*), w którym sieć rekurencyjna otrzymuje sekwencję o długości k stempli czasowych, każdy zawierający dane nieprzetworzone lub wektor cech, zaś jako wynik przetwarzania (odpowiedź klasyfikatora) analizowana jest tylko odpowiedź sieci z ostatniego (k -tego) stempla czasowego.

We wstępnie przeprowadzonych testach potwierdzono, że podstawowe komórki pamięci RNN nie są odpowiednie do rozpoznawania długich, silnie zróżnicowanych co do długości (dla różnych klas) sekwencji dlatego rozważono użycie komórek pamięci z bramkami. Sesje treningowe z komórką pamięci GRU osiągały zbieżność dla mniejszej liczby epok, a obliczenia zajęły mniej czasu niż dla bardziej złożonych komórek LSTM, zwłaszcza dla dużej liczby epok (ponad 500). Z tego powodu w badaniach skupiono się na uczeniu z wykorzystaniem komórki GRU. Implementacja modelu RNN została opracowana przy użyciu bibliotek TensorFlow i Scikit-learn [161] i uruchomiona na stacji roboczej Nvidia DGX Station wyposażonej w 4 wysoce zoptymalizowane pod kątem uczenia maszynowego karty graficzne Tesla V100. Kod napisany na potrzeby utworzenia modelu sieci RNN wykonany w języku Python udostępniony został w serwisie GitHub³.

W celu optymalizacji wartości hiperparametrów modelu wykorzystano podejście losowego poszukiwania (ang. *random search*). Dla każdego z trzech typów zbiorów danych przeprowadzono po 128 sesji treningowych, każda o osobniczym zestawie wartości hiperparametrów, które były losowane z następujących zbiorów:

- tempo uczenia, jednorodnie z zakresu $\langle 1E-7, 0,05 \rangle$,
- ilość warstw n_{LAY} sieci, jednorodnie ze zbioru [1, 2, 3, 4, 5],
- ilość neuronów n_{NEU} (ta sama dla każdej warstwy), jednorodnie z zakresu $\langle 10, 100 \rangle$
- funkcja aktywacji warstw ukrytych, jednorodnie ze zbioru [tanh, softsign, ReLU, ELU, SELU [162]],
- prawdopodobieństwo zatrzymania połączenia za warstwą (ang. *dropout output keep probability*), jednorodnie ze zbioru [0,5; 0,7; 0,7; 1],
- inicjalizer jądra (ang. *kernel initializer*) komórki GRU, jednorodnie ze zbioru [inicjalizer He, inicjalizer Xaviera, żaden].

Dla parametru odrzucania (ang. *dropout*), który jest jedną z metod regularyzacji sieci neuronowych, rozważono większe prawdopodobieństwo wylosowania wartości 0,7, gdyż w pewnych badaniach, w których wyniki zależały od wartości hiperparametrów wartość ta okazała się efektywna [138].

³ Kod modelu RNN dostępny na: <https://github.com/ChrisQlasty/RNNforGestures>

Po wylosowaniu wartości poszczególnych hiperparametrów zastosowano następujący zestaw reguł i limitów, celem uczynienia obliczeń bardziej efektywnymi czasowo:

- jeżeli ($n_{LAY} == 1$), `wykonaj_dropout = Fałsz` (dla jednowarstwowych sieci odrzucanie może pogarszać zdolność uczenia się sieci),
- dopóki ($n_{NEU} \cdot n_{LAY} > 100$), $n_{NEU} -= 3$ (ograniczenie wielkości sieci),
- jeżeli (`funkcja_aktywacji == ReLU`), skaluj wektory / cechy wejściowe do sieci do zakresu $\langle 0, 1 \rangle$, lub $\langle -1, 1 \rangle$ w przeciwnym wypadku (ponieważ zakres wartości pozostałych funkcji aktywacji mieści się w przedziale $\langle -1, 1 \rangle$ a `ReLU` $\langle 0, 1 \rangle$).

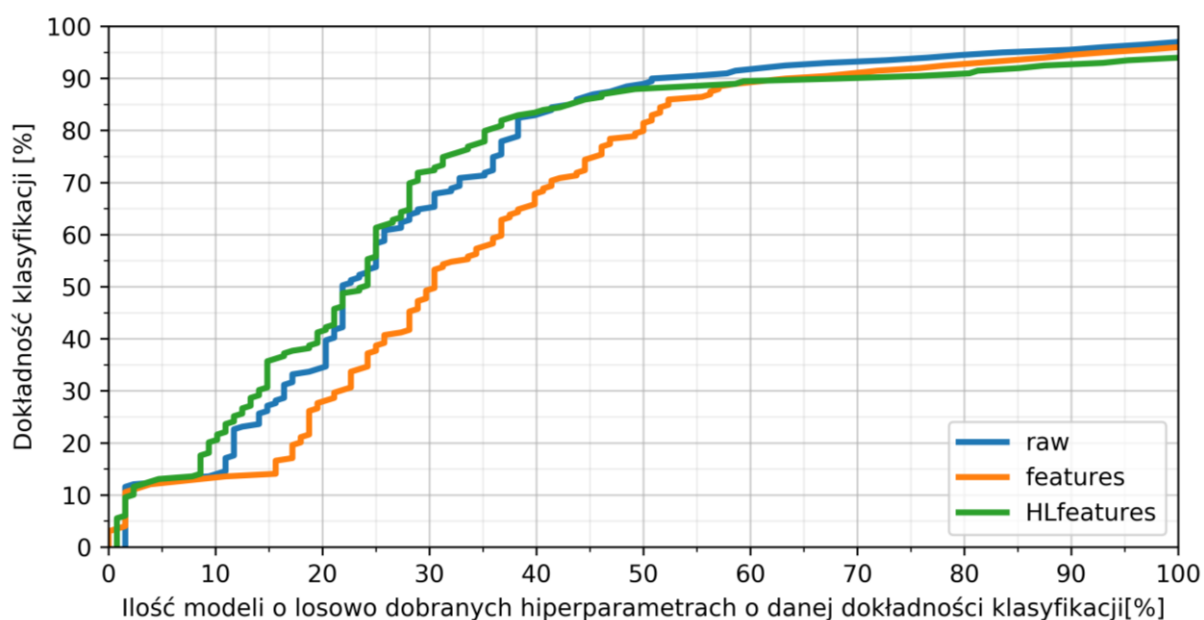
Warunki wczesnego zatrzymania treningu (ang. *early stopping*):

- zatrzymaj sesję treningową jeżeli na przestrzeni 100 epok nie zanotowano poprawy dokładności klasyfikacji na zbiorze walidacyjnym,
- zatrzymaj sesję treningową w epoce e jeżeli najwyższą dotychczasową dokładność klasyfikacji na zbiorze walidacyjnym zaobserwowano przed epoką $e/2$ [111],
- zatrzymaj sesję treningową jeżeli dokładność klasyfikacji na zbiorze walidacyjnym wynosi 0% na przestrzeni 4 kolejnych sprawdzeń, gdyż nastąpił najprawdopodobniej wybuch lub zanik gradientu (ang. *exploding / vanishing gradient*),
- w pozostałych przypadkach, trenuj sieci przez minimum 100 i maksimum 1000 epok.

Można zauważyć, że limit całkowitej liczby neuronów prowadzi do badań sieci neuronowych o topologiach płytkich lecz szerokich (np. $n_{LAY}=1$ i $n_{NEU}=100$) oraz głębokich lecz wąskich (np. $n_{LAY}=5$ i $n_{NEU}=20$). W badaniach zastosowano adaptacyjny algorytm optymalizacyjny Adam, który implementuje zmienne tempo uczenia sieci neuronowej, i który oparty jest na uczeniu sieci za pomocą wielu rozłącznych podzbiorów zbioru treningowego, tzw. minipakietów (ang. *minibatch*) [163]. Wybrano niewielki rozmiar minipakietu, 256, ponieważ kosztem dłuższego czasu obliczeń i większej liczby potrzebnych epok wytrenowane w ten sposób modele posiadają lepszą zdolność do uogólniania [164]. Ponadto w celu poprawienia generalizacji zastosowano losowanie minipakietów w każdej epoce (ang. *minibatch shuffling*). Zastosowano także przycinanie gradientów (ang. *gradient clipping*) z normą równą 1 dla wszystkich zbiorów danych. Jako wyjście sieci zastosowano gęstą warstwę (ang. *dense, fully connected*) z 27 neuronami (co odpowiada liczbie gestów do rozpoznania) i funkcją aktywacji *softmax*. Interwał ewaluacji rezultatów uczenia (sprawdzanie klasyfikatora na zbiorze walidacyjnym) wyniósł 5 epok.

4.4.3. Wyniki ewaluacji metod rozpoznawania gestów

Wyniki dokładności klasyfikacji gestów na zbiorze walidacyjnym, przez modele trenowane na trzech typach reprezentacji gestu, zaprezentowano w postaci wykresów skumulowanych wartości (Rys. 4.24). Spośród 128 losowych treningów 50% modeli wyszkolonych na danych nieprzetworzonych uzyskało dokładności klasyfikacji na zbiorze walidacyjnym na poziomie co najmniej 90%. Najlepszy z nich osiągnął wynik walidacji na poziomie 96,86%. Ponad 35% modeli wyszkolonych na wektorach podstawowych cech osiągnęło wynik walidacji na poziomie 90%, podczas gdy najskuteczniejszy z nich osiągnął 95,95%. W przypadku wektorów cech wysokiego poziomu jako wejść do sieci neuronowej, niewiele powyżej 30% modeli przekracza 90% dokładności klasyfikacji, zaś najdokładniejszy model osiąga 93,78%.



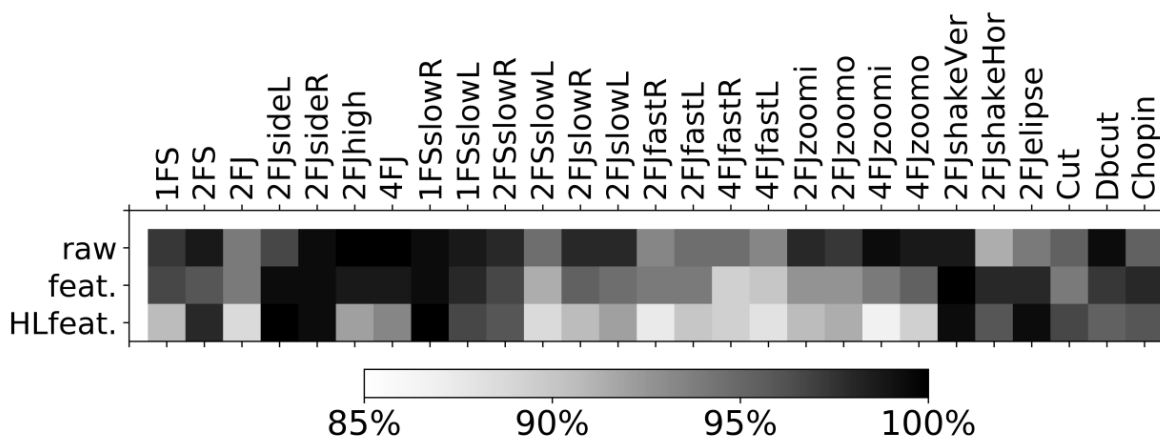
Rys. 4.24 Wykresy skumulowanych dokładności klasyfikacji gestu na zbiorze walidacyjnym, przez modele o wylosowanych hiperparametrach, dla trzech typów reprezentacji gestu.

Odchylenie standardowe dokładności klasyfikacji w obrębie trzech najlepszych modeli wyszkolonych na każdym z typów reprezentacji danych w każdym przypadku było niższe niż 0,08%, co oznacza, że mają one bardzo zbliżoną dokładność. Dlatego spośród nich wybrano modele trenowane przez największą liczbę epok jako modele, które będą ewaluowane na zbiorach testowych. Podsumowanie wybranych modeli z wylosowanym zestawem hiperparametrów i ich szczegółową wydajnością na zbiorze testowym przedstawia Tabela 4.8.

Tabela 4.8 Podsumowanie wybranych modeli dla każdego typu reprezentacji danych.

reprezentacja danych:	nieprzetworzone	wektory cech	wektory cech HL
n_{inputs}	8	6	3
n_{NEU}	97	48	32
n_{LAY}	1	2	3
Tempo uczenia	0,0027	0,0051	0,0036
Odrzucanie	1,0	0,7	0,7
Funkcja aktywacji warstw ukrytych	tanh	softsign	ELU
Inicjalizer jądra	Xavier	He	Żaden
Epoka treningowa z najw. dokł.	280	520	245
Dokł. na zbiorze walidacyjnym	96,69%	95,83%	93,75%
Dokł. na zbiorze testowym	96,89%	95,75%	93,38%
Min. dokł. klasy na zb. testowym	91,33%	89,33%	86,67%
Med. dokł. klasy na zb. testowym	98%	96%	92%
Max. dokł. klasy na zb. testowym	100%	100%	100%
SD dokł. klas na zb. testowym	2,39%	2,90%	4,26%

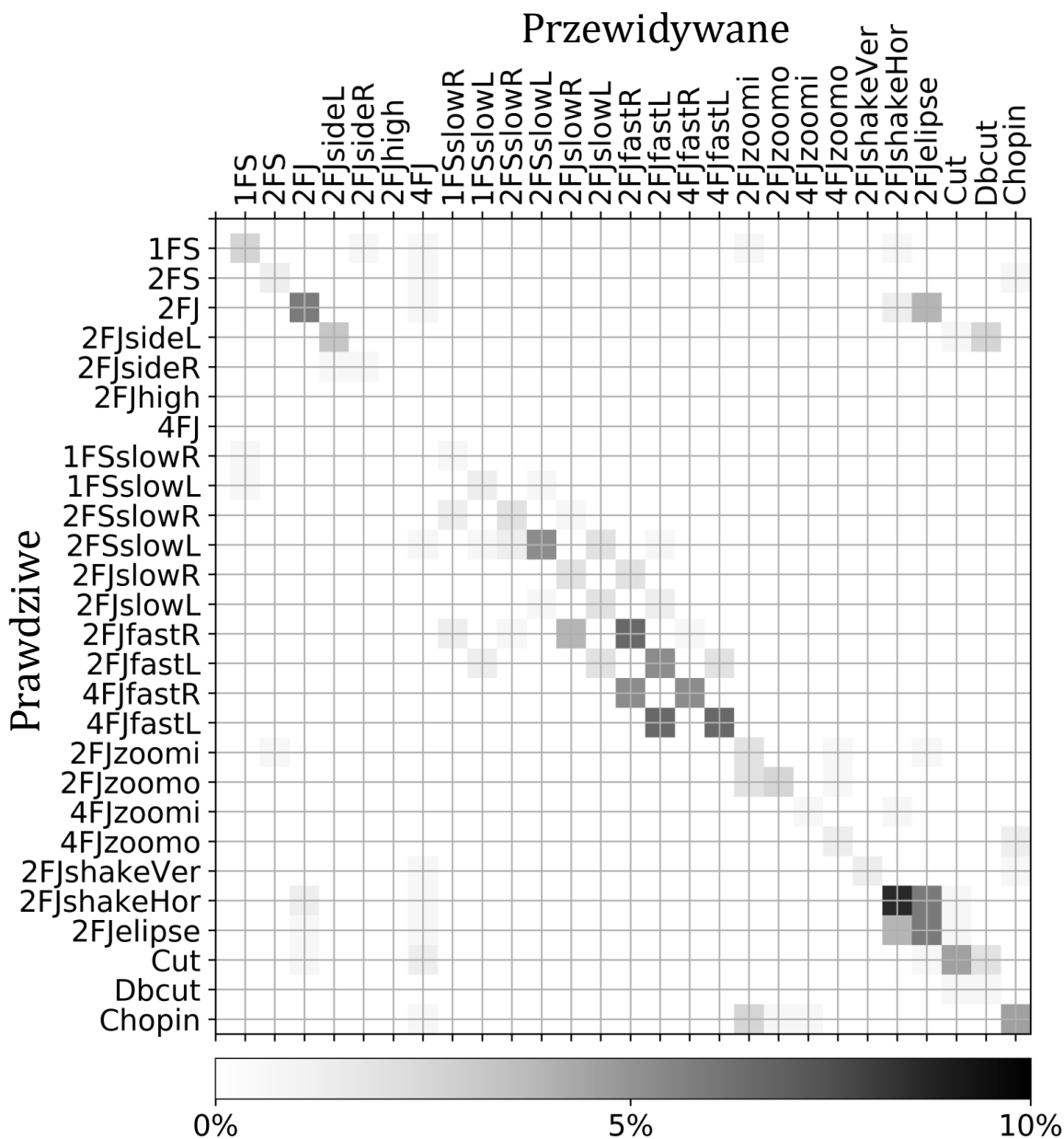
Dokładność wykrywania poszczególnych gestów z zestawu testowego przez każdy z wybranych modeli została przedstawiona na Rys. 4.25.



Rys. 4.25 Dokładność detekcji gestów, sprawdzona na zbiorze testowym, dla wybranych modeli dla każdego typu reprezentacji gestu.

Na Rys. 4.25 przedstawiono zmodyfikowaną tablicę pomyłek dla najlepszego modelu, wyszkolonego na nieprzetworzonych danych i ewaluowanego na zbiorze testowym, który osiąga 96,89% dokładności. Modyfikacja klasycznej tablicy pomyłek polega na tym, że przekątna w zaprezentowanej macierzy została uzyskana poprzez odjęcie wartości oryginalnej tablicy pomyłek od 100%. W związku z tym, suma wszystkich wartości

w każdym rzędzie, poza wartością z diagonalnej, sumują się do wartości na przekątnej. Dzięki temu, zakres mapy kolorów z obrazu pozwala wizualnie uwydatnić błędy klasyfikatora, których poziom jest skorelowany ze stopniem zacielenia komórki w macierzy (większa dynamika obrazu).



Rys. 4.26 Zmodyfikowana tablica pomyłek wybranego modelu rozpoznającego gesty z danych nieprzetworzonych. Skala kolorów wskazuje na procentowy błąd rozpoznania danego gestu.

Z wyników zaprezentowanych na Rys. 4.24 widać, że statystycznie nieznacznie łatwiej jest wytrenować model RNN na wektorach cech wysokiego poziomu niż na nieprzetworzonych danych i osiągnąć wysoką dokładność klasyfikacji (do 80%). W tym momencie liczba modeli przeszkolonych na obu tych zestawach danych przekracza

liczbę modeli przeszkolonych na danych typu wektory zwykłych cech o ponad 10 punktów procentowych. Jednak w miarę jak poszczególne sesje treningowe stają się głębsze (gdzie dokładność klasyfikacji przekracza 90%), model oparty na wektorach cech HL przestaje być konkurencyjny. Jest to najprawdopodobniej spowodowane faktem, że chociaż w tym zestawie danych poza jest już zidentyfikowana dla modelu RNN, to wykorzystany klasyfikator pozy (który dostarcza jedną z trzech cech wejściowych dla tych modeli) obciążony jest swoim własnym błędem klasyfikacji, który obciąża ogólną zdolność rozpoznawania gestów przez RNN. Nieprzetworzony zbiór danych jest jedynym, w którym ponad połowa próbnych zestawów hiperparametrów ustanowiła modele zdolne do klasyfikacji gestów z dokładnością ponad 90%. Dokładność najlepszego modelu przeszkolonego w zakresie danych nieprzetworzonych przewyższała najlepsze modele wyszkolone na wektorach cech. Mimo, że model danych nieprzetworzonych nie był wyszkolony dla największej liczby epok, charakteryzuje go najniższe odchylenie standardowe dokładności klasyfikacji poszczególnych klas, 2,39% (im mniej, tym lepiej), a najwyższa wartość najgorszej zaobserwowanej dokładności dla dowolnego z gestów z analizowanego zestawu wynosi 91,33% (im więcej, tym lepiej).

Dodatkowo, z pierwszej dziesiątki (top10) modeli wyszkolonych na danych nieprzetworzonych, 7 składało się z 3 lub większej liczby warstw ukrytych. Z top10 modeli wyszkolonych na wektorach cech tylko 3 modele charakteryzowały się liczbą ukrytych warstw większą lub równą 3. Dla cech HL, połowa z top10 modeli miała co najmniej 3 warstwy. Pokazuje to, że kiedy sieć otrzymała już wyodrębnione informacje (cechy), preferowała ona płytkie topologie. Natomiast w przypadku danych nieprzetworzonych głębokie sieci okazały się być bardziej konkurencyjne pod względem zdolności do skutecznego uczenia się. Z drugiej strony modele wybrane z top3 dla każdego typu reprezentacji danych nie potwierdzają tej obserwacji wykonanej na większej próbie. Tabela 4.8 pokazuje również, że najlepsze modele preferowały dużą liczbę neuronów w sieci, która była bliska lub równa zastosowanemu limitowi (100). W związku z tym, w przyszłości mogą zostać rozważone modele o większej ilości parametrów (neuronów), zdolne do nauczenia się większej liczby zależności w dostarczonym zbiorze danych.

Dokładności rozpoznawania gestów statycznych osiągniętej przez wybrany model wyszkolony na zestawie danych *HLfeatures* nie należy bezpośrednio porównywać z dokładnością opracowanego klasyfikatora pozy dłoni [156]. Można jednak zaobserwować,

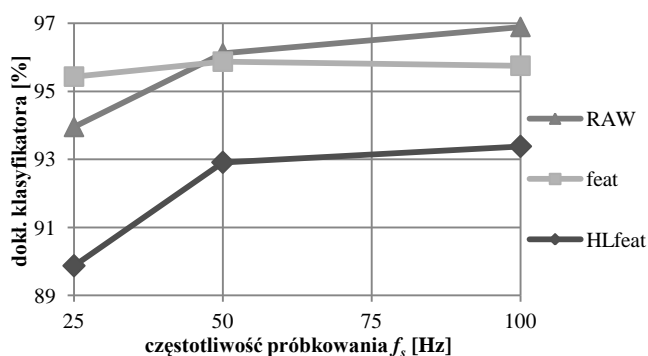
że model RNN wykrywał pozy 2FJ ze stosunkowo podobną (± 1 p.p.) dokładnością. Rozpoznawanie pozy 1FS było o 7p.p. mniej efektywne, podczas gdy dla pozy 4FJ było wyższe o około 8p.p. Jednak model wyszkolony z użyciem nieprzetworzonego zestawu danych okazał się być w stanie rozpoznawać pozy statyczne znacznie lepiej. Biorąc pod uwagę nieprzetworzony zestaw danych, wykrywanie gestów przesunięcia dłoni (*statyczna poza i ścieżka*) najczęściej zawodziło w aspekcie rozpoznawania właściwych póz dłoni, a także w różnicowaniu jej prędkości poruszania się (Rys. 4.26). Gest Cut (*poza dynamiczna*) był nieznacznie mylony z gestami podwójnego cięcia Dbcut (Rys. 4.26). Gest elipsy najczęściej mylony był z horyzontalnym potrząsaniem, a 2FJ z elipsą (Rys. 4.26). Może to wynikać z prędkości wykonywanych gestów, dlatego w przyszłych pracach należy rozważyć dodatkowe filtrowanie czasoprzestrzenne ramek danych gestu. Polegać ono może na nie dodawaniu do sekwencji kolejnej ramki danych, jeśli poza dłoni nie przemieści się o dostateczną odległość względem poprzednio uzyskanej pozycji (*COG, max*).

Ogólna dokładność klasyfikacji wybranej sieci neuronowej wytrenowanej na danych nieprzetworzonych jest na wysokim poziomie (96,89%), co można porównać do prac innych autorów opisujących działanie podobnego typu czujników (np. 94,8%, uśredniając po wszystkich konfiguracjach [65]). Jednak w przypadku czujników o niskiej rozdzielczości większość metod zaproponowanych przez innych autorów rozpoznaje wyraźnie mniejszą liczbą gestów. Dlatego opisany czujnik o budowie liniowej w połączeniu z modelem opartym na RNN z komórkami pamięci GRU, który jest w stanie śledzić zarówno pozę ręki, jak i jej trajektorię z dobrą dokładnością stanowi konkurencyjne rozwiązanie.

Rozmiar sieci neuronowych (całkowita liczba neuronów w warstwach ukrytych) wybranych modeli wytrenowanych dla poszczególnych typów danych jest bardzo podobny (97, 98 i 96 neuronów). Dlatego wykorzystanie wektorów cech lub cech HL (mniejsza liczba wejść do modelu) nie dawałoby szczególnie istotnej przewagi w stosunku do danych nieprzetworzonych, biorąc pod uwagę czas przejścia w przód (*forward pass*) podczas inferencji. Z powodu zaś dodatkowego czasu potrzebnego do obliczenia poszczególnych cech rozwiązania te mogą być nawet bardziej kosztowne czasowo. Wysoką dokładność modelu można potencjalnie podnieść choćby zwalniając sieć od obowiązku rozpoznawania tempa ruchu (szybkość przesunięcia dłonią). W przyszłych pracach w modelu rozpoznawania gestów bazującym na RNN może być również brany pod uwagę rozmiar dłoni osoby korzystającej z czujnika. Parametr szerokości dłoni mógłby zostać dodany do obciążeń

komórek pamięci w pierwszym stemple czasowym jako informacja stacjonarna w modelu dla sekwencji bazującym na sieciach rekurencyjnych co w interpretacji obrazów przeprowadził m.in. Karpathy i in. [165]. Dalsze badania mogą odnosić się także do bardziej wydajnych poszukiwań hiperparametrów w oparciu o jedną z technik optymalizacji [166] i zwiększoną liczbę losowanych modeli.

Należy również zwrócić uwagę, na fakt, iż przygotowywanie danych do postaci cech do modelu RNN nie było wykonywane w czasie rzeczywistym przez mikrokontroler lecz po zapisie, na komputerze. Opracowanie danych przez mikrokontroler do postaci np. *HLfeatures* jest zaś ograniczone przede wszystkim przez tempo przetwarzania ramki danych przez dwuwarstwową sieć neuronową zaimplementowaną w obrębie pamięci jednostki obliczeniowej czujnika optycznego (50Hz)(rozdział 4.2.5). W związku z tym, dla wybranych topologii przeznaczonych do każdego z trzech rodzajów stopnia przetworzenia danych (Tabela 4.8) przeprowadzono dodatkowe sesje treningowe na sekwencjach o niższych częstotliwościach próbkowania, możliwych do uzyskania na mikrokontrolerze bez względu na rodzaj przetwarzania danych. Na oryginalnych sekwencjach przeprowadzono podpróbkowanie (ang. *downsampling*) pozostawiając co 2 i co 4 próbkę, uzyskując efektywne częstotliwości próbkowania na poziomie 50 i 25Hz. Wyniki ewaluacji wytrenowanych sieci na zbiorach testowych zostały przedstawione na Rys. 4.27 oraz w formie tabeli, podsumowującej także zapotrzebowanie energetyczne czujnika (Tabela 4.9). Szacowane zużycie prądu przez elementy optoelektroniczne czujnika, I_{TK} , w zależności od częstotliwości próbkowania zostało obliczone jak zostało to opisane w sekcji 3.5.6 Pobór prądu i kalibracja oraz w publikacji autora rozprawy [157].



Rys. 4.27 Przebiegi dokładności klasyfikatorów gestów dla różnego typu danych i dla różnych częstotliwości próbkowania.

Tabela 4.9 Dokładność klasyfikatorów gestów dla różnych częstotliwości próbkowania f_s i różnego typu danych oraz zużycie prądu przez czujnik.

f_s [Hz]	RAW [%]	feat [%]	HLfeat [%]	I_{TK} [mA]
25	93,95	95,43	89,87	1,24
50	96,12	95,87	92,91	2,48
100	96,89	95,75	93,38	4,96



Zaprezentowane wyniki wskazują na niższą dokładność klasyfikatorów bazujących na nieprzetworzonych danych i na cechach wysokiego poziomu dla częstotliwości próbkowania 25Hz. Model trenowany na danych z cech podstawowych w badanym zakresie utrzymuje wysoką dokładność klasyfikacji (ponad 95%). Jednak wobec badań, których wyniki przedstawiono w sekcji 3.5.5 widać iż dla częstotliwości próbkowania nieco poniżej 30Hz możliwe jest wykrywanie jedynie połowy przesunięć dłonią ($C=53\%$), zaś dla 50Hz jest to ok. 85% przesunięć. Rozróżnianie prędkości przesunięcia dłonią jest ważną cechą z punktu widzenia intuicyjnego znaczenia poszczególnych gestów (np. przełączanie o 3 lub 10 kontrolki w zależności od wolnego lub prędkiego przesunięcia) stąd nie zalecane jest wobec zaproponowanej konstrukcji czujnika stosowanie częstotliwości próbkowania poniżej 40Hz ($C=75\%$). Omówiony klasyfikator bazujący na modelu rekurencyjnych sieci neuronowych, wytrenowany przykładowo z użyciem nieprzetworzonych danych próbkowanych z częstotliwością 50Hz stanowić może realizację bloku funkcjonalnego *Analiza sekwencji fazy Rozpoznawanie gestu* systemu rozpoznawania gestów GRS. Implementacja modelu o tej złożoności i wykonanego za pomocą biblioteki TensorFlow nie jest możliwa w obrębie mikroprocesora PIC24FV16KA302. Możliwa jest jednak ewaluacja takiego modelu na urządzeniach mobilnych z systemem Android lub iOS stosując jedno z rozwiązań: TensorFlow for Mobile lub TensorFlow Lite.

W trakcie treningu sieci rekurencyjnej dla zgromadzonego zbioru danych czas obliczeń przez komputer przenośny z kartą graficzną Nvidia GTX 960M potrzebny na wykonanie 100 epok trwał w przybliżeniu 1 godzinę. Nieduża optymalizacja modelu rekurencyjnych sieci neuronowych pod kątem treningu na kartach graficznych w metodach biblioteki TensorFlow spowodowała, iż czas treningu na komputerze Nvidia DGX Station, przy wykorzystaniu tego samego skryptu, był praktycznie taki sam. Jednak duża pojemność pamięci każdej z czterech kart komputera DGX (16 GB) i charakter treningu (duża liczba niezależnych prób) pozwoliły na równoczesne uruchomienie 8 skryptów (po 2 na kartę graficzną) wykonujących treningi dla różnych topologii sieci czyli na ośmiokrotne przyspieszenie prac związanych z poszukiwaniem optymalnej topologii sieci.

4.5. Podsumowanie rozdziału

W pierwszej części rozdziału przedstawiono schemat systemu rozpoznawania gestów GRS wraz z opisem jego bloków, których funkcjonalność uszczegółowiono na przestrzeni analizy wyników poszczególnych eksperymentów zaprezentowanych w niniejszym rozdziale. Następnie porównano ze sobą dwie metody rozpoznawania póż dłoni dla zbioru danych zgromadzonego przez czujnik operujący w trybie aktywnym. Wykazano, że model bazujący na sztucznych sieciach neuronowych cechuje się wyższą dokładnością klasyfikacji póż zwartych (1FS, 2FJ, 4FJ), niż metoda dopasowywania krzywych obserwacji do wzorców (90,02% do 81,19%). Topologię danej sieci neuronowej wybrano ze względu na wysoką dokładność klasyfikacji oraz niewielkie rozmiary (4 cechy, 5 neuronów w 1 WU, 2 neurony w 2 WU), gdyż w tych badaniach minimalizowano także złożoność obliczeniową metod. Metoda II pozwoliła także na znacznie szybsze przetwarzanie ramek danych w stosunku do Metody I (50Hz vs 10Hz). Z tych powodów, w dalszych badaniach związanych z wykrywaniem póż dłoni stosowano Metodę II. Wobec wysokiej dokładności klasyfikatora trzech póż zwartych oraz wyników zaprezentowanych w rozdziale 3, wskazujących na możliwość rozróżnienia przez czujnik liniowy odpowiednio rozpostartych palców dłoni możliwe jest udowodnienie części tezy I odnoszącej się do minimalnej ilości (4) rozróżnianych póż dłoni przez czujnik o liniowej budowie, wyposażony w przesłony optyczne.

W dalszej kolejności zaprojektowano i przeprowadzono serię eksperymentów wskazujących na możliwość określenia warunków świetlnych otoczenia wystarczających dla utrzymania wysokiej dokładności detekcji póż danego trybu pracy czujnika. Ponadto wykazano, iż jasne i ciemne warunki oświetlenia wymagają wytrenowania osobnych klasyfikatorów, nauczonych na danych zgromadzonych w trakcie pracy w każdym z trybów, aby osiągnąć dokładność klasyfikacji na poziomie ponad 90%. Dzięki temu osiągnięto odpowiednio 93,46% dla trybu aktywnego i 98,76% dokładności rozpoznawania trzech póż zwartych dla trybu pasywnego. Zastosowanie trybu pasywnego pozwala na zaoszczędzenie 93,34% energii potrzebnej przez elementy optoelektroniczne sensora na pracę w trybie aktywnym, niezależnie od przyjętej częstotliwości próbkowania.

Przedstawione wnioski pozwalają na stwierdzenie o udowodnieniu drugiej z tez postawionych w niniejszej rozprawie doktorskiej:

Opracowanie dwóch trybów pracy optycznego czujnika o liniowej budowie z zastosowaniem (tryb aktywny) lub bez zastosowania (tryb pasywny) dodatkowego oświetlenia w podczerwieni oraz opracowanie dedykowanych im metod przetwarzania danych umożliwia uzyskanie wyższej niż 90% dokładności rozpoznawania pól dłoni przy minimalizacji energii dla potrzeb zastosowań mobilnych.

W trzeciej części rozdziału wykorzystano metodę losowego poszukiwania hiperparametrów modelu i zweryfikowano po 128 topologii rekurencyjnych sieci neuronowych do każdego typu danych (nieprzetworzone ramki danych, w postaci cech oraz w postaci cech przetworzonych). Wyniki wskazują na brak konieczności przetwarzania poszczególnych ramek danych kiedy wykorzystana jest analiza sekwencji o odpowiedniej topologii. Wykazano, iż dla $f_s \geq 50\text{Hz}$ dokładność klasyfikacji gestów przez model wytrenowany na nieprzetworzonych pakietach jest wyższa (96,86%), niż dla modeli uczonych na cechach niższego i wyższego rzędu (odpowiednio 95,95% i 93,78%). Ponadto redukcja częstotliwości próbkowania o połowę pozwala na utrzymanie wysokiej dokładności klasyfikacji gestów przy jednoczesnym dwukrotnym zmniejszeniu zapotrzebowania na zużycie prądu przez czujnik. W uczeniu sieci neuronowych wykorzystano bazę danych, którą tworzyło 27 klas gestów, znacznie więcej niż obsługują opisywane w literaturze czujniki optyczne podobnego typu. Gest rozpoznawany z najniższą dokładnością przez najlepszy z modeli wykrywany był z częstotliwością 91,33%. Można zatem orzec, iż każdy ze zdefiniowanych gestów był rozpoznawany z wysoką dokładnością. Powyższe stwierdzenia pozwalają na pozytywne zweryfikowanie trzeciej z tez postawionych w rozprawie.

Opracowanie modelu bazującego na rekurencyjnej sieci neuronowej wykorzystującego jako dane wejściowe nieprzetworzone pakiety danych z optycznego czujnika o liniowej budowie pozwala na klasyfikację co najmniej 20 gestów dłoni z dokładnością nie mniejszą niż przy wykorzystaniu danych przetworzonych w formie wektora cech z rozpoznaniem pozami dłoni.

Przedstawione zdolności czujnika optycznego o liniowej budowie do rozpoznawania gestów i jego niskie zużycie energii pozwalają na rozważenie kilkudetektorowych czujników jako istotnego, energooszczędnego wsparcia lub nawet jako alternatywy dla interfejsów opartych na analizie wideo, szczególnie w kontekście zastosowania do urządzeń mobilnych jak multimedialne okulary czy telefony.

5. PRAKTYCZNE ZASTOSOWANIA CZUJNIKA OPTYCZNEGO O LINIOWEJ BUDOWIE I WNIOSKI KOŃCOWE

W tym rozdziale w ramach podsumowania opracowanych w niniejszej rozprawie doktorskiej rozwiązań oraz możliwości dalszego rozwoju czujnika optycznego o liniowej budowie opisane zostały jego praktyczne zastosowania. Rozdział zawiera także podsumowanie niniejszej rozprawy doktorskiej.

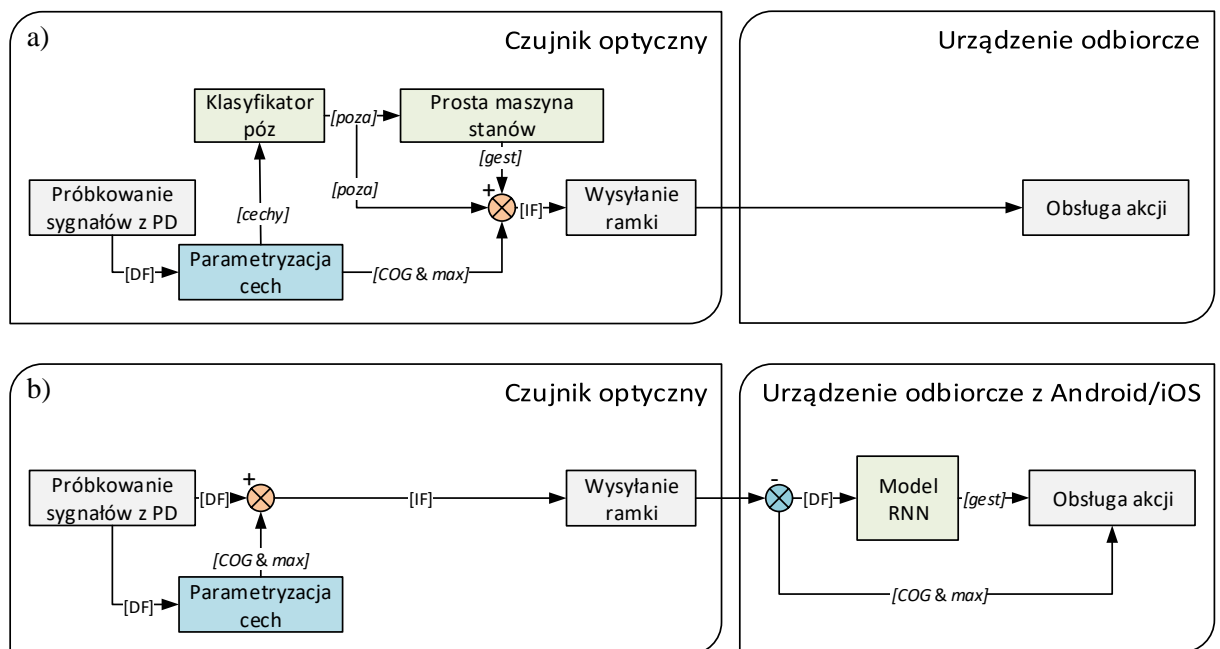
5.1. Podsumowanie opracowanych rozwiązań

Opracowany czujnik gestów, w zależności od zastosowanych metod przetwarzania danych i możliwości urządzenia odbiorczego, może zostać wykorzystany w komunikacji z obsługiwanym urządzeniem na kilka sposobów. Jedną z możliwości jest dostarczanie przez czujnik z wysoką częstotliwością ($f_s=50\text{Hz}$) ramki informacyjnej (ang. *information frame*, IF) zawierającej szereg podstawowych danych na temat pozy dłoni widzianej w danym cyklu próbkowania: jej lokalizację względem czujnika poprzez wartości parametrów *COG* i *max* oraz kod rozpoznanej pozy statycznej. Dodatkowo możliwe jest także wysłanie kodu rozpoznanego prostego gestu (np. przytrzymanie dłoni w pozie 2FJ, przesunięcie dłoni

w pozie 4FJ) uzyskanego dzięki prostej maszynie stanów implementowanej w obrębie mikrokontrolera czujnika optycznego, co zweryfikowano w ramach przeprowadzonych badań [5][167]. Dzięki temu czujnik jest w stanie dostarczać danych zarówno na temat prostych gestów dyskretnych jak i informacji użytecznych przy definicji gestów ciągłych, których interpretacja odbywać się będzie na platformie odbiorczej. Metodę tę wykorzystano podczas testów interakcji przedstawionych w punkcie 5.2.

Inną możliwością jest przesyłanie z czujnika gestów nieprzetworzonych ramek danych (lub danych w postaci cech profilu natężenia światła) do urządzenia mobilnego, zdolnego do przeprowadzenia inferencji (np. posiadającego system Android). W ten sposób, dzięki podaniu zarejestrowanej przez czujnik sekwencji na wejście do zapisanego na zewnętrznym urządzeniu modelu rekurencyjnych sieci neuronowych możliwe jest sterowanie urządzeniem za pomocą dużej liczby złożonych gestów dyskretnych. Ważne jest również, aby nadsyłana z czujnika ramka informacyjna, IF, zawierała wiadomość o lokalizacji pozy, lub aby urządzenie odbiorcze obliczało parametry *COG* i *max* na podstawie nieprzetworzonych danych celem zachowania możliwości obsługi gestów ciągłych.

Obydwie z opisanych możliwości wykorzystania optycznego czujnika gestów do różnego rodzaju urządzeń docelowych zaprezentowano schematycznie na Rys. 5.1.



Rys. 5.1 a) Sterowanie urządzeniem przez czujnik optyczny wysyłający ramkę informacyjną w postaci lokalizacji pozy, kodu pozy i rodzaju gestu z prostej maszyny stanów. b) sterowanie urządzeniem przez czujnik wysyłający lokalizację pozy oraz nieprzetworzoną ramkę danych, którą urządzenie odbiorcze zinterpretuje poprzez model RNN.

W kontekście badań Murthy'ego rozważono wymagania, które powinien spełniać projektowany czujnik gestów [31]. Wymaganie odporności osiągnięto dzięki zastosowaniu adaptacyjnego trybu pracy: pasywnego lub aktywnego w zależności od warunków oświetlenia czujnika. Zastosowanie sztucznych sieci neuronowych i wykorzystanie mikrokontrolera PIC24FV16KA302 pozwoliło na klasyfikację pów dłoni z częstotliwością 50Hz co pozwala spełnić wymaganie efektywności obliczeniowej przy rozpoznawaniu pów. Rozpoznawanie gestów z użyciem zaprojektowanego i wytrenowanego modelu RNN wymaga jednak użycia systemu operacyjnego Android na zewnętrznym urządzeniu (np. okulary multimedialne lub komputer jednopłytkowy - ang. *single board computer*, SBC) co może spowolnić czas reakcji czujnika. Na spełnienie wymagania tolerancji użytkownika wskazuje osiągnięta wysoka dokładność metod klasyfikacji pów (sekcje 4.2.5 i 4.3) i gestów dłoni (sekcja 4.4.3) trenowanych i testowanych na zbiorach utworzonych z udziałem grupy ochotników. Wymaganie skalowalności czujnika rozważyć można w kontekście jego zdolności do rozpoznawania zarówno gestów dyskretnych jak i ciągłych co czyni go elastycznym w zależności od typu obsługiwanego interfejsu czy aplikacji.

Biorąc pod uwagę dodatkowe wymagania stawiane interfejsom przez Chenga i in. [32] należy odnieść się do: rozpoznawania w czasie rzeczywistym, minimalizacji fałszywych alarmów i rozpoznawania fazy rozpoczęcia i zakończenia gestu. Pierwsze z wymagań powiązane jest z czasem przetwarzania. Według US Department of Defense Design Criteria Standard - Human Engineering czas reakcji systemu krótszy niż 200ms pozwala na odbiór interakcji z interfejsem graficznym użytkownika jako płynnej [168]. Dostarczanie przez czujnik informacji o typie i lokalizacji pozy z okresem 20ms (50Hz) pozwala na spełnienie tego kryterium. Minimalizacja fałszywych alarmów w rozważanym czujniku o budowie liniowej następuje przez zastosowanie progu odchylenia standardowego ramki danych. Podobnie rozpoznawanie fazy początku i zakończenia gestu, w zależności od trybu pracy, zależy od przekroczenia przez odchylenie standardowe ramki danych konkretnej wartości progu.

W kontekście wymagań sprecyzowanych przez autora rozprawy w punkcie 3.1 stwierdzono, iż: kształt i rozmiar czujnika pozwalają na umieszczenie go w urządzeniach mobilnych, co zostało zweryfikowane przez integrację czujnika w obrębie oprawek okularów platformy eGlasses (sekcja 5.2.2). Ponadto, zaprojektowany czujnik optyczny, zdolny do rozpoznawania gestów, składa się z kilku detektorów, nie zaś z przetwornika

matrycowego. Zasięg czujnika w trybie aktywnym sięga, zgodnie z założeniami, ok. 5 cm. Podjęta została minimalizacja zużycia energii przez czujnik poprzez oszacowanie wymaganej częstotliwości próbkowania i zastosowanie różnych trybów pracy sensora.

Zaprezentowana w niniejszej rozprawie metoda przełączania między pasywnym i aktywnym trybem pracy czujnika optycznego o liniowej budowie polega na oszacowaniu świetlnych warunków otoczenia mierzonych przez dany czujnik. Określenie warunków odbywa się przez zestawienie wartości maksymalnej ramki danych z wyznaczoną wartością progową T_{max} (0,6 V). W przypadku adaptacji opisywanej metody przełączania przez inne urządzenia, konieczne jest przeprowadzenie analogicznych badań i wykonanie kalibracji. Podobnie, jako inne stałe wymagające kalibracji wyróżniono próg minimalnego odchylenia standardowego ramki danych, powyżej którego analizowana jest poza dłoni zarówno dla trybu pasywnego sd_{Tp} (0,13 V) jak i aktywnego sd_{Ta} (0,10 V). Otrzymane wartości progów mogą się bowiem różnić w przypadku urządzeń o innym rozłożeniu elementów optoelektronicznych lub wykorzystujących inne elementy wykazujące się różną czułością (fotodiody) czy mocą (diody LED).

5.2. Opracowane przykłady zastosowań

W wyniku prac związanych z badaniem metod bezdotykowej interakcji w ramach projektu eGlasses opracowano dwa rozwiązania konstrukcyjne dla optycznego czujnika gestów: czujnik jako moduł dołączany np. do zewnętrznego komputera jednopłytkowego oraz czujnik wbudowany w multimedialne okulary.

5.2.1. Czujnik optyczny jako dołączany moduł

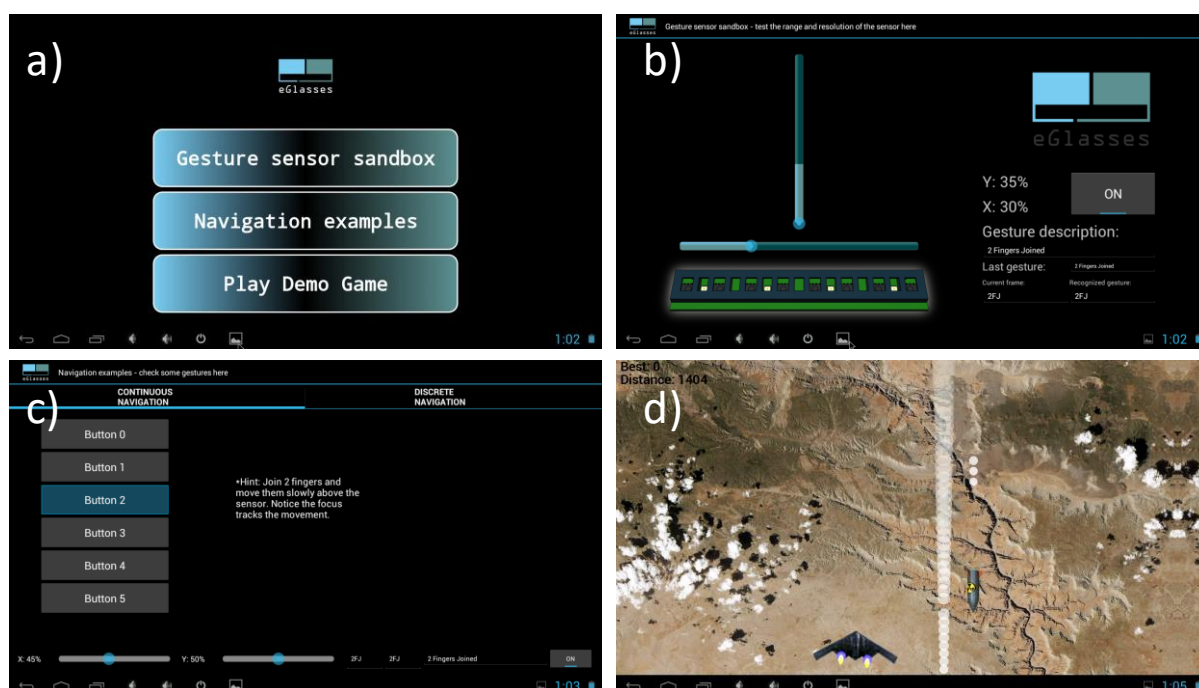
W ramach badań nad metodami interakcji z wykorzystaniem optycznego czujnika gestów wykonana została wersja czujnika wyposażona w złączkę wyprowadzającą na długich przewodach poszczególne piny czujnika (zasilanie V_{DD} , napięcie odniesienia V_{REF} , masę GND, odbiór i wysyłanie danych za pomocą UART: Rx, Tx). Dzięki temu, optyczny czujnik gestów został połączony z urządzeniem odbiorczym w postaci komputera jednopłytkowego Odroid-X (procesor Samsung Exynos4412) firmy Hardkernel przez jego piny GPIO (ang. *general-purpose input/output*). Mikrokomputer Odroid-X posiada także porty HDMI oraz port USB, dzięki którym możliwe jest podłączenie do niego monitora oraz również klasycznych interfejsów takich jak klawiatura czy mysz.

Zastosowane urządzenie odbiorcze pracowało pod systemem operacyjnym Android. Czujnik gestów komunikował się z mikrokomputerem Odroid-X wykorzystując port szeregowy UART i wysyłał dane gestu z częstotliwością 50Hz (Rys. 5.2a). Sensor skonfigurowano tak, iż w każdej ramce wyjściowej wysyłał on do aplikacji cztery pola: kod rozpoznanej przez sztuczne sieci neuronowe pozy, kod prostego gestu (poza statyczna lub przesunięcie danej pozy, w danym tempie i kierunku) uzyskany przez prosty układ maszyn stanów [5][167] oraz parametry *COG* i *max* ramki danych wskazujące na lokalizację dłoni w stosunku do czujnika.

W celu zaprezentowania części możliwości optycznego czujnika gestów o liniowej budowie opracowana została w języku Java dedykowana aplikacja na system Android 4.2. Testowy program zawiera kilka opcji (Rys. 5.3a). Jedną z nich jest okno ćwiczeń (piaskownica, ang. *sandbox*), w którym użytkownik ma możliwość sprawdzić zasięg czujnika oraz jego zdolność do lokalizacji ręki w stosunku do urządzenia. Stopień zbliżenia dłoni do czujnika oraz stopień jej wychylenia względem środka czujnika zaobserwować można przez poziom wypełnienia odpowiednich kontrolki interfejsu graficznego użytkownika (pasków ładowania, ang. *seek bar*)(Rys. 5.3b). W tym oknie wyświetlane są również informacje związane z wykonanymi prostymi gestami dyskretnymi (statycznymi i dynamicznymi). Aplikacja demonstracyjna prezentuje również dwie z możliwych metod nawigacji po elementach interfejsu graficznego użytkownika: ciągłą (Rys. 5.3c) oraz dyskretną. W trybie ciągłym przesunięcie dłoni wzdłuż czujnika z początku aż do końca jego zakresu widoczności powoduje podświetlanie aktywności kolejnych ikonki programu od pierwszej (lewy górny róg) do ostatniej (prawy dolny róg ekranu). W trybie dyskretnym (uruchamianym przez przełączanie zakładki) należy wykonać pełny gest przemieszczenia dłoni wzdłuż czujnika aby przełączyć aktywną ikonkę o jedną do przodu (ruch w prawo) lub do tyłu (ruch w lewo). Ponadto użytkownik otrzymuje możliwość weryfikacji precyzji odwzorowania przemieszczania dłoni przez czujnik na ruchy samolotu na ekranie podczas gry polegającej na unikaniu rakiet (Rys. 5.3d). Ponieważ cały cykl interpretacji danych odbywał się w tym wypadku w obrębie mikrokontrolera czujnika gestów, urządzenie współpracujące z nim (Odroid-X) nie musiało zużywać swoich zasobów obliczeniowych na analizę nieprzetworzonych danych. Rys. 5.2b prezentuje dłoń nad czujnikiem w trakcie sterowania ruchem samolotu na tle ekranu z uruchomioną grą.



Rys. 5.2 a) Połączenie czujnika optycznego z komputerem jednopłytkowym Odroid-X (przewody: zasilający V_{DD} , napięcia odniesienia V_{REF} , masy GND oraz odbioru i wysyłania danych Rx, Tx). b) Dłoń sterująca ruchem samolotu za pomocą optycznego czujnika gestów podłączonego do urządzenia Odroid-X.



Rys. 5.3 Kolejne ekrany aplikacji demonstrującej działanie i możliwości podczerwonego czujnika optycznego, napisanej dla systemu Android.

5.2.2. Czujnik optyczny wbudowany w okulary multimedialne

Optyczny czujnik gestów testowano także w obudowie okularów multimedialnych, które powstały w ramach europejskiego projektu *Interaktywne okulary dla mobilnego i percepcyjnego przetwarzania (eGlasses)*. Dzięki swojej podłużnej budowie swobodnie mieści się on w bocznej części oprawki okularów (Rys. 5.4a). Czujnik został podłączony do portów cyfrowych platformy eGlasses wyposażonej w procesor TI OMAP 4460 z wykorzystaniem portu szeregowego UART. Sensor gestów skonfigurowany był analogicznie jak w testach z wykorzystaniem mikrokomputera Odroid-X, zaś okulary multimedialne również pracowały pod tą samą wersją systemu operacyjnego Android (4.2).

Dzięki wyposażeniu platformy okularów w zintegrowany wyświetlacz NTE dla prawego oka ich użytkownik widzi informacje wyświetlane przez system operacyjny (lub aplikację) niezależnie od ruchów głowy. Oprócz opisanej w poprzednim podpunkcie dedykowanej aplikacji do testowania czujnika gestów w tym rozwiązaniu konstrukcyjnym czujnik użyto także do sterowania ruchami drona kołowego (Rys. 5.4b, Rys. 5.5). Poprzez pokazywanie naprzeciwko czujnika optycznego dyskretnych gestów dłoni, wyzwalane były różnego rodzaju akcje (np. obrót drona, podskok, przejazd do przodu, przejazd do tyłu), które wysyłane były do drona przez platformę eGlasses za pomocą standardu komunikacji bezprzewodowej WiFi. Wykorzystanie zdalnie sterowanego za pomocą gestów drona posiadającego kamerę i transmitującego obraz do osoby sterującej nim może być przykładowo wykorzystane jako wsparcie do nadzoru bezpieczeństwa osób starszych i chorych (Rys. 5.5).



Rys. 5.4 a) Zastosowanie optycznego czujnika gestów o liniowej budowie w okularach eGlasses. Czarna część na tle białej oprawki to bloki optyczne czujnika gestów. b) Użytkownik okularów wykonujący gest 2FS wyzwalający jedną z akcji zachowania drona kołowego.



Rys. 5.5. Zdalny monitoring zdarzeń w pomieszczeniach jako praktyczny przykład sterowania dronem za pomocą czujnika gestów umieszczonego w okularach multimedialnych. Dolna część rysunku zawiera podgląd obrazu widzianego z drona, który może być transmitowany do okularów.

Działanie optycznego czujnika o liniowej budowie w obu przedstawionych rozwiązaniach konstrukcyjnych a także z użyciem m.in. opisanej aplikacji testowej na system Android przedstawiano podczas szeregu międzynarodowych wydarzeń:

- Active and Assisted Living (AAL) Forum, Ghent, Holandia, 2015, prezentacja z zakresu multimedialnych okularów i technik interakcji człowiek-komputer we współpracy z iHomeLab, Szwajcaria,
- Active and Assisted Living (AAL) Forum, St. Gallen, Szwajcaria, 2016, prezentacja z zakresu multimedialnych okularów i technik interakcji człowiek-komputer we współpracy z iHomeLab, Szwajcaria,
- UbiComp 2016 (*Conference on Pervasive and Ubiquitous Computing*) w Heidelbergu, w Niemczech, stanowisko z demonstracjami metod interakcji z wykorzystaniem czujnika optycznego i aplikacji testowej dla systemu Android,
- CHIST-ERA Seminar and Exhibition, multimedialne okulary i techniki interakcji człowiek-komputer (prezentacja i stanowisko z demonstracjami), Bruksela, Belgia, 2017,
- Global Internet of Things Summit (GIoTS) Genewa, Szwajcaria, 2017, prezentacja z zakresu multimedialnych okularów i technik interakcji człowiek-komputer we współpracy z iHomeLab, Szwajcaria.

Ponadto na stronie projektu eGlasses zamieszczony został film prezentujący część możliwości zaprojektowanego i wykonanego optycznego czujnika gestów⁴.

5.3. Przyszłe prace

W ramach dalszego rozwoju możliwości i rozważenia potencjalnych dziedzin wykorzystania opisywanego w niniejszej rozprawie doktorskiej czujnika gestów wskazane jest przeprowadzenie szeregu dodatkowych badań i opisanie jego praktycznych zastosowań.

5.3.1. Badania nad praktycznym użytkowaniem czujnika gestów

W niniejszej rozprawie doktorskiej nie zostały rozważone praktyczne odwzorowania gestów na akcje wywoływane w graficznym interfejsie użytkownika wybranej aplikacji (komputerowej bądź urządzenia mobilnego) z reprezentatywną grupą użytkowników końcowych. Praktyczne zastosowanie opracowanego czujnika wymagałoby szczegółowego i wieloaspektowego badania z udziałem użytkowników końcowych z uwzględnieniem np. różnych warunków i dziedziny zastosowania. Wyniki takich badań mogą powstać na podstawie analizy ocen użytkowników wystawionych w formie odpowiedzi na pytania ankietowe np. typu zamkniętego (np. tak/nie/nie wiem) bądź ze skalą np. Likerta (np. zdecydowanie się zgadzam, raczej się zgadzam, nie mam zdania itd.).

Badania powinny być przeprowadzone dla konkretnej aplikacji, w scenariuszach użytkowania różniących się np. stopniem oświetlenia pomieszczenia (np. brak światła słonecznego / silne światło słoneczne), wykorzystanym rozwiązaniem konstrukcyjnym czujnika (czujnik w okularach multimedialnych / czujnik jako moduł umieszczony naprzeciwko użytkownika jak np. w kiosku informacyjnym), typem zadań do wykonania (np. nawigacja po menu / obsługa zdjęć / gra zręcznościowa w przechodzenie labiryntu) czy okolicznościami wykorzystania determinującymi np. swobodę zachowania użytkownika, potrzebne tempo oraz precyzję działania (np. warunki sali operacyjnej / podróż autobusem).

Dzięki wykorzystaniu modelu rekurencyjnych sieci neuronowych osiągnięto wysoką dokładność rozpoznawania 27 gestów przez czujnik optyczny. Liczba klasyfikowanych gestów umożliwia zatem przypisanie każdej z akcji potrzebnych zdaniem autora rozprawy do wygodnej nawigacji urządzenia mobilnego, zdefiniowanych w podpunkcie 2.6,

⁴ <http://www.e-glasses.net/?p=297>

osobnego gestu. Propozycję przypisania, gestów do akcji, wykonaną możliwie z uwzględnieniem znaczenia metaforycznego gestów zawiera Tabela 5.1.

Tabela 5.1 Przypisanie gestów rozpoznawanych przez czujnik optyczny do akcji wyzwalanych w aplikacji mobilnej.

	Kategoria	Akcja	Gest
1	Trans- formacja	obrót w prawo	2FS slow R
2		obrót w lewo	2FS slow L
3		zwiększ	4FJ zoom in
4		zmniejsz	4FJ zoom out
5	Symulacja	odtwarzaj / kontynuuj	4FJ
6		pauza	2FS
7		stop	2FJ shake hor
8		zwiększ prędkość	2FJ side R
9		zmniejsz prędkość	2FJ side L
10		wycisz	2FJ high
11		zwiększ głośność	2FJ zoom in
12		zmniejsz głośność	2FJ zoom out
13	Wybieranie	następna kontrolka	1FS slow R
14		poprzednia kontrolka	1FS slow L
15		zaznacz wszystko	2FJ shake hor
16		3 kontrolki w przód	2FJ slow R
17		3 kontrolki w tył	2FJ slow L
18		10 kontrolki w przód	2FJ fast R
19		10 kontrolki do tyłu	2FJ fast L
20		następna karta	4FJ fast R
21		poprzednia karta	4FJ fast L
22	Menu	otwórz pop-up menu	Chopin
23		wybierz / potwierdź	2FJ
24		zamknij/cofnij/anuluj	Cut
25		odśwież	Elipsa
26		opuść aplikację	Dbcut

W ramach weryfikacji praktycznego zastosowania czujnika optycznego w kontekście zaproponowanego przypisania gestów do wyzwalanych akcji należy wykonać testy z wykorzystaniem reprezentatywnej grupy ochotników. Mogą one polegać na realizowaniu zadań z każdej kategorii zdefiniowanych akcji (*transformacja*, *symulacja*, *wybieranie* i *menu*) według ustalonego scenariusza, np. „obrót zdjęcie dwa razy w prawo i powiększ”, „zwiększ głośność wideo, następnie je wycisz, zwiększ prędkość wideo, zapauzuj, kontynuuj i zastopuj”, „przełącz widok aplikacji wybierając następną kartę, zmień zaznaczenie aktywnej

kontrolki na dwunastą kontrolkę w przód po czym na dziewiątą do tyłu, przełącz jedną kartę w tył i zaznacz wszystkie kontrolki”, „otwórz menu pop-up, wybierz domyślną opcję, wyjdź z menu a następnie wyjdź z aplikacji”. W przypadku testów takiego typu należy zanotować czas potrzebny użytkownikowi na wykonanie testu, potrzebną ilość prób, liczbę pomyłek użytkownika, liczbę pomyłek modelu rozpoznawania gestów i ogólną ocenę stopnia wygody interakcji użytkownika

Ważnym czynnikiem sprawdzenia możliwości praktycznego zastosowania optycznego czujnika gestów jest także przeprowadzenie badań związanych z inferencją. Wytrenowany model RNN należy wyeksportować do urządzenia mobilnego z systemem Android i osadzić w napisanej do celów badawczych aplikacji. Aplikacja taka powinna podawać na wejście opracowanego modelu pojedyncze sekwencje (gesty) ze zbioru testowego i mierzyć czas potrzebny sieci neuronowej na wykorzystanym urządzeniu (o danym procesorze) na obliczenie odpowiedzi modelu (inferencja). Uzyskany średni czas obliczeń w stosunku do długości sekwencji wejściowej pozwoliłby określić czy czas oczekiwania na odpowiedź sieci na gesty o maksymalnej długości jest wystarczająco krótki aby reakcja systemu była postrzegana jako płynna (np. poniżej 0,2s).

5.3.2. Możliwości dalszego zastosowania i rozwoju

Czujnik optyczny o liniowej budowie w postaci takiej jak zaprezentowano w niniejszej rozprawie doktorskiej oprócz zastosowania w okularach multimedialnych może zostać rozważony także jako dodatkowy interfejs do telefonu, tabletu lub kiosku informacyjnego. Zaprezentowane testy czujnika z użyciem dedykowanej aplikacji na system Android wskazują, iż może on znaleźć użycie także w grach zręcznościowych czy w aplikacjach wymagających połączenia obsługi gestów ciągłych i dyskretnych.

Możliwości jakie niesie połączenie funkcjonalności gestów ciągłych i dyskretnych są znaczące dla precyzji i wygody w trakcie obsługi urządzenia mobilnego. Przykładowo ustalanie wartości wybieranej przez kontrolkę suwaka (ang. *slide bar* albo *seek bar*) może odbywać się w dwóch trybach. Tryb normalny może polegać na tym, że kiedy dłoń ułożona jest w pozę np. 1FS i wykonuje zakres ruchów w osi x czujnika, jej położenie (wartość parametru COG) bezwzględnie determinuje położenie suwaka na ekranie (np. maksymalne wychylenie dłoni na lewo od środka czujnika, póki znajduje się ona w jego polu widzenia powoduje ustawienie suwaka na 0% wartości; analogicznie maksymalne

wychylenie w prawo ustawia suwak na 100%). Tryb wysokiej precyzji może oznaczać, że dla dłoni ułożonej w innej pozie (np. 2FJ) ruchy dłoni względem czujnika będą oznaczać zmiany relatywne położenia suwaka o ograniczonym zakresie (np. maksymalne wychylenie dłoni na lewo od środka czujnika, póki znajduje się ona w jego polu widzenia spowoduje ustawienie suwaka na wartości o 5p.p. mniejszą niż początkowo ustawiona, przed pojawieniem się pozy).

Kolejną z możliwości rozszerzenia funkcjonalności optycznego czujnika o liniowej budowie jest użycie dwóch czujników np. po obydwu stronach okularów multimedialnych lub gogli VR. W ten sposób możliwa będzie np. znacznie wygodniejsza nawigacja w programach do wizualizacji 3D. W takim przypadku jeden z czujników służyłby do blokowania danej osi (np. przez pozę 1FS osi x , 2FJ osi y , 2FS osi z) zaś gesty ciągle wykonywane na drugim czujniku (np. powolne zbliżanie i oddalanie pozy 2FJ) mogłyby wykonywać rotację widoku względem unieruchomionej osi.

5.3.3. Praktyczne możliwe zastosowania w ochronie zdrowia

Liczna baza rozpoznawanych gestów dłoni oraz cecha bezkontaktowości pozwala rozważyć czujnik optyczny jako interfejs stosowany w sytuacjach codziennych czy w przemyśle jak np. przez serwisantów maszyn, mechaników, monterów urządzeń elektronicznych, kucharzy, podczas prac domowych i w ogrodzie itp. Ponadto cechy te czynią interfejs szczególnie praktycznym także w miejscach publicznych jak stadiony, galerie handlowe, teatry i wszędzie tam gdzie możliwe jest wyeliminowanie dotykowej obsługi np. kiosków informacyjnych, biletomatów, włączników świateł itp. celem zachowania higieny i redukcji ryzyka przenoszenia bakterii, grzybów i wirusów [20][21].

Cecha bezkontaktowości pełni kluczową rolę w ochronie zdrowia gdzie zachowanie sterylności dłoni jest szczególnie ważne zwłaszcza w sytuacjach wymagających naruszenia powłok ciała (np. operacje, pobór krwi). Możliwość zaś wyzwania pojedynczym gestem dużej liczby różnych akcji powoduje, że dostęp do wielu funkcji aplikacji współpracującej z czujnikiem optycznym może być szybki i precyzyjny co jest istotnym czynnikiem np. w trakcie przeprowadzania zabiegów przeszczepu organu.

Klasycznym przykładem zastosowania interfejsów bezdotykowych na sali operacyjnej jest przeglądanie przez lekarza zdjęć dokumentacji medycznej pacjenta [12][13]. Tabela 5.1 zawiera propozycje przypisania gestów do wywoływanych reakcji interfejsu graficznego użytkownika. W kontekście kategorii *transformacja* opisano 4 akcje możliwe

do wykorzystania podczas przeglądania zdjęć (obrót zdjęcia w prawo/lewo za pomocą wolnego przesunięcia poży 2FS w prawo/lewo; zwiększanie i zmniejszanie zdjęcia za pomocą gestów zbliżania i oddalania poży 4FJ). Dodatkowo przełączanie do następnego / poprzedniego zdjęcia może się odbywać za pomocą gestów wolnego przesunięcia poży 1FS w prawo / lewo (kategoria *wybieranie*).

Rozważyć można również użycie czujnika gestów w okularach multimedialnych noszonych przez lekarza w trakcie zabiegu do kontroli parametrów sali operacyjnej. Za pomocą gestu Cut lekarz mógłby zmieniać aktualnie sterowane źródło światła (np. jedną z kilku lamp oświetlających operowaną część ciała) zaś za pomocą gestu ciągłego (np. odpowiednie przybliżanie dłoni o pozie 4FJ) mógłby dostosować precyzyjnie poziom jego jasności według własnych preferencji. Przykładowo za pomocą gestu elipsy może być także na sali operacyjnej załączana klimatyzacja.

W podobny sposób jak do sterowania światłem, z wykorzystaniem czujnika optycznego, w trakcie wykonywania zabiegu lekarz mógłby zachowując sterylność rąk płynnie dostosowywać poziomy np. jasności ekranu w wyświetlaczu okularów multimedialnych, kontrastu wyświetlanego zdjęcia lub dynamiki obrazu przez zmianę zakresu wyświetlanych kolorów [15](zmiana parametru może następować przez pokazanie gestu 2FS zaś zmiana jego wartości przez gest ciągły np. stopień bliskości poży 4FJ w stosunku do czujnika).

Za pomocą odpowiednio wyposażonych okularów multimedialnych możliwe jest także potencjalnie przeprowadzenie szkolenia [16] czy wykonanie konsultacji przez chirurga w trakcie wielogodzinnego zabiegu gdyż bezkontaktowy interfejs czujnika optycznego wykorzystać można do nawiązania połączenia (np. gest Chopin wyzwała ukryte menu, z którego gestami przesunięcia poży 1FS w lewo i prawo można wybrać odpowiednią osobę w książce telefonicznej i gestem 2FJ potwierdzić nawiązanie połączenia).

Na zautomatyzowanych oddziałach szpitalnych gdzie pacjenci podłączeni są do monitorów parametrów życiowych (np. poziomu ciśnienia krwi, tętna, poziomu saturacji krwi tlenem itp.) możliwe jest wprowadzenie bezdotykowego modułu obsługi urządzeń. W trakcie przeprowadzanego obchodu lekarz mający styczność z wieloma pacjentami i sprzętami do nich podłączonymi mógłby przeglądać ich aktualne i archiwalne dane medyczne bez konieczności dotykania przedmiotów strefy pacjenta. Otwieranie historii wyników mogłoby odbywać się za pomocą gestu 2FS. Przełączanie zaś wyświetlania kolejnych parametrów za pomocą gestów przesunięcia 1FS w lewo i w prawo.

Za pomocą gestów zbliżania i oddalania poży 4FJ możliwe byłoby zmienianie skali czasowej pokazywanego parametru (np. wykres średniego ciśnienia w ciągu ostatniej godziny / w ostatnich 24 godzinach / w ostatnim tygodniu itp.).

Tak jak w środowisku szpitalnym każdy kontakt dłoni z obiektem jest zagrożeniem dla jej sterylności tak w miejscach użytku publicznego to skala (duża ilość osób obsługujących jedno urządzenie) powoduje realne zagrożenie zarażeniem. Obsługa urządzeń informacyjnych bądź porządkujących kolejkowanie w miejscach użytku publicznego także może zostać praktycznie zrealizowana za pomocą bezdotykowego czujnika optycznego. W biletomacie w urzędach czy w punktach pobrań w laboratorium często obecnych jest kilka typów okienek obsługi klienta (np. wydawanie dowodów rejestracyjnych, składanie podania o nowy dowód itp.) lub opcji dla pacjenta (np. odbiór wyników, rejestracja, informacja). W takim wypadku do wskazania wybranej opcji możliwe jest pokazanie jednej z pól dłoni (np. 1FS – odbiór wyników, 2FS – rejestracja, 4FJ – informacja).

W galeriach handlowych czy muzeach, tam gdzie kioski informacyjne mogą prezentować mapę danego miejsca, możliwe jest bezdotykowe przeglądanie topologii budynku. Za pomocą gestów np. wolnego przesunięcia 2FS w prawo lub w lewo graficzny interfejs użytkownika mógłby zmieniać aktualnie wyświetlaną kondygnację budynku. Wykorzystując gesty ciągłe i np. zbliżanie poży 4FJ do czujnika optycznego pozwoli na zmianę skali wyświetlanego planu zaś poruszanie poży 2FJ w osiach prostopadłej i równoległej do sensora umożliwić może przesuwanie widoku w prawo / lewo oraz w górę/dół.

Powszechnie obecne w przestrzeni publicznej są także automaty z napojami oraz produktami spożywczymi, często stojące na dworcach i w innych miejscach masowo odwiedzanych przez ludzi potencjalnie przenoszących choroby zakaźne. Zastosowanie optycznego czujnika bezdotykowych gestów jako opcję umożliwiającą wybranie produktu zmniejszy ryzyko zarażenia chorobą przez interfejs dotykowy. Wybór odpowiednich produktów mógłby odbywać się z wykorzystaniem gestów ciągłych – zbliżanie poży 2FJ w stosunku do czujnika może ustalać kolumnę, zaś przesuwanie poży 2FJ w prawo/lewo może zmieniać rząd wybieranego produktu w tabelarycznym układzie automatu. Wykonanie gestu np. elipsy anuluje akcję zaś gestu Dbcut pozwoli użytkownikowi zaakceptować wybrany produkt i przystąpić do zapłaty transakcją np. z wykorzystaniem karty zbliżeniowej.

5.4. Podsumowanie rozprawy doktorskiej

Wobec stopniowego upowszechniania się urządzeń z segmentu elektroniki ubieralnej, zwłaszcza okularów multimedialnych, oraz wobec atrakcyjności możliwych dziedzin ich zastosowania, rozwój bezdotykowych interfejsów gestów jest tematyką aktualną i rozwojową. Możliwości modułu rozpoznawania gestów w stosunku do jego efektywności energetycznej i obliczeniowej są zaś szczególnie istotne w zastosowaniu do urządzeń mobilnych.

Na podstawie przeglądu rozwiązań literaturowych prostych czujników gestów oraz rodzajów interfejsów stosowanych w komercyjnych urządzeniach typu multimedialne okulary stwierdzono, iż brakuje rozwiązań bezdotykowych czujników kilkudetektorowych zdolnych do rozpoznawania dużej liczby gestów dłoni (ponad 20). Podana liczba wynika z przeprowadzonej analizy ilości gestów wymaganych do wygodnej i precyzyjnej nawigacji po systemie mobilnym wykorzystując jedynie prosty czujnik gestów, nie bazujący na matrycy (np. kamera) a więc mniej wymagający energetycznie i obliczeniowo.

W niniejszej rozprawie doktorskiej zaproponowano użycie liniowej konstrukcji kilkudetektorowego czujnika optycznego, o odpowiednio dobranych parametrach fizycznych i technicznych, jako interfejsu gestów dłoni. Zebranie danych dotyczących wielkości palców i poczucia szybkości przesunięcia dłonią przez grupę ochotników wykonano celem utworzenia statystyki, której wartości brano pod uwagę przy projektowaniu wymagań rozpoznawania różnych aspektów gestu przez czujnik. Na podstawie symulacji opartych na metodzie Monte Carlo oraz serii eksperymentów wykonanych z użyciem fizycznej implementacji czujnika zweryfikowano różne konfiguracje rozłożenia elementów optoelektronicznych sensora w takich parametrach jak między innymi: spłaszczenie profilu podświetlenia, zdolność rozdzielcza czy czułość przestrzenna i czasowa. W dalszej części badań wskazano również na możliwości wykorzystania czujnika liniowego jako interfejsu gestów ciągłych. Następnie, porównano ze sobą dwie metody rozpoznawania pól zwartych dłoni na podstawie danych z czujnika liniowego pracującego w trybie aktywnym. Wykazano, iż Metoda II, bazująca na sztucznych sieciach neuronowych, jest zdolna do klasyfikacji pól z wyższą dokładnością niż Metoda I, polegająca na dopasowaniu wzorca i obserwacji (90,02% w stosunku do 81,19%). Z tego względu, to klasyfikator wykorzystujący sieci neuronowe brano pod uwagę w dalszych badaniach. W kontekście funkcji adaptacji czujnika rozważono wpływ zróżnicowanego oświetlenia na wartości jego parametrów wyjściowych, używanych do rozpoznawania gestów (poza dłoni,

lokalizacja względem sensora). Przeprowadzone badania pozwoliły na zdefiniowanie wartości progu natężenia światła otoczenia, po przekroczeniu którego czujnik powinien przełączyć swój tryb pracy na pasywny lub aktywny. Wskazano także, iż wysoką dokładność klasyfikacji zarówno w warunkach silnego światła (93,46%) jak i jego pomijalnego poziomu (98,76%) utrzymać można dzięki zastosowaniu osobnych klasyfikatorów póż dla każdego z dwóch trybów pracy. Zastosowanie pasywnego trybu pracy pozwala zaś na redukcję poboru energii przez czujnik o 93,34%. Następnie porównano dokładność klasyfikacji gestów, czyli całych sekwencji zlokalizowanych póż, wykorzystując zbiory danych o różnym stopniu przetworzenia (nieprzetworzone dane, dane w postaci cech, dane w postaci przetworzonych cech). Zaprezentowano, iż zastosowanie modelu rekurencyjnych sieci neuronowych pozwala na rozpoznawanie dużej bazy gestów dłoni (27 gestów) z najwyższą dokładnością kiedy jako wejście do modelu wykorzystane zostaną dane nieprzetworzone (niemal 97%). Wskazano także, iż obniżenie częstotliwości próbkowania ze 100Hz do 50Hz, pozwalające na dwukrotną redukcję prądu pobieranego przez elementy optoelektroniczne czujnika (z 4,96 mA do 2,48 mA), zwiększa błąd klasyfikacji jedynie o 0,77p.p. dla danych nieprzetworzonych.

Na podstawie przeprowadzonych w niniejszej rozprawie badań i analiz stwierdzono, iż osiągnięto główny cel rozprawy, którym było: *zaprojektowanie metod rozpoznawania gestów dłoni i sposobów interakcji z urządzeniami za pomocą kilkudetektorowego optycznego czujnika o liniowej budowie oraz cele szczegółowe.*

W częściach podsumowujących poszczególne rozdziały wskazano na wyniki badań pozwalające na udowodnienie sformułowanych w ramach niniejszej rozprawy tez. W szczególności w sekcjach 3.5.4 oraz 4.2.5 zaprezentowano wyniki pozwalające na udowodnienie tezy:

- I. *Możliwe jest opracowanie kilkuelementowego czujnika optycznego zbudowanego z liniowo rozłożonych fotodiod oraz diod LED, jak również przesłon optycznych formujących pole widzenia fotodiod, umożliwiającego rozpoznawanie co najmniej 4 póż dłoni.*

W sekcjach 4.2.5, 4.3.4 i 0 omówiono badania wskazujące na udowodnienie tezy:

- II. *Opracowanie dwóch trybów pracy optycznego czujnika o liniowej budowie z zastosowaniem (tryb aktywny) lub bez zastosowania (tryb pasywny) dodatkowego oświetlenia w podczerwieni oraz opracowanie dedykowanych im metod przetwarzania danych umożliwia uzyskanie wyższej niż 90% dokładności rozpoznawania pól dłoni przy minimalizacji energii dla potrzeb zastosowań mobilnych.*

W sekcjach 4.4.1 i 4.4.3 zarysowano charakter problemu i przedstawiono wyniki pozwalające na udowodnienie tezy:

- III. *Opracowanie modelu bazującego na rekurencyjnej sieci neuronowej wykorzystującego jako dane wejściowe nieprzetworzone pakiety danych z optycznego czujnika o liniowej budowie pozwala na klasyfikację co najmniej 20 gestów dłoni z dokładnością nie mniejszą niż przy wykorzystaniu danych przetworzonych w formie wektora cech z rozpoznanymi pozami dłoni.*

5.5. Oryginalne osiągnięcia

Wynikiem badań nad udowodnieniem tez sformułowanych w niniejszej rozprawie doktorskiej są następujące oryginalne osiągnięcia autora rozprawy:

- opracowanie krytycznej analizy stanu wiedzy w zakresie tematyki rozprawy (sekcja 1),
- opracowanie zbioru akcji do wygodnego sterowania urządzeniami mobilnymi (sekcja 2.6),
- zaprojektowanie i realizacja czujnika optycznego o liniowej budowie [5][47][107][157][167][169],
- opracowanie, realizacja i eksperymentalna weryfikacja metod automatycznego rozpoznawania pól dłoni [154][156][169],
- opracowanie, realizacja i weryfikacja metody przełączania trybów pracy optycznego czujnika gestów [158],
- opracowanie i eksperymentalna weryfikacja modelu symulacyjnego zachowania optycznego czujnika o liniowej budowie [154][157],

- przeprowadzenie analizy porównawczej zdolności modelu RNN do klasyfikacji gestów na podstawie danych o różnym stopniu przetworzenia z optycznego czujnika gestów (sekcja 4.4.3) [169],
- opracowanie modelu pozwalającego, na podstawie danych z kilkudetektorowego czujnika gestów, na rozpoznawanie znacznie większej liczby gestów (27, z możliwością dalszego powiększenia), niż inne podobnego typu czujniki optyczne opisane w literaturze (sekcje 1 i 4.4.3) [169],
- opracowanie dwóch sposobów wykorzystania czujnika optycznego, w zależności od możliwości urządzenia odbiorczego, polegających na wysłaniu nieprzetworzonych danych do modelu RNN w odbiorniku lub wysłaniu pakietu opisującego proste gesty (sekcja 5.1).

Podziękowania

Badania przeprowadzone w ramach niniejszej rozprawy doktorskiej
współfinansowane były ze środków uzyskanych w projekcie
Interaktywne okulary dla mobilnego i percepcyjnego przetwarzania (eGlasses)
oraz ze środków statutowych
Wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej.

BIBLIOGRAFIA

- [1] J. Lumsden and S. A. Brewster, "A Paradigm Shift: Alternative Interaction Techniques for Use with Mobile & Wearable Devices," *CASCON '03 Proc. 2003 Conf. Cent. Adv. Stud. Collab. Res.*, pp. 197–210, 2003.
- [2] S. Brewster, J. Lumsden, M. Bell, M. Hall, and S. Tasker, "Multimodal 'eyes-free' interaction techniques for wearable devices," *Proc. Conf. Hum. factors Comput. Syst. - CHI '03*, no. 5, p. 473, 2003.
- [3] E. Freeman, S. Brewster, and V. Lantz, "Towards usable and acceptable above-device interactions," *Proc. 16th Int. Conf. Human-computer Interact. with Mob. devices Serv. - MobileHCI '14*, pp. 459–464, 2014.
- [4] M. Serrano, B. Ens, and P. Irani, "Exploring the Use of Hand - To - Face Input for Interacting with Head - Worn Displays," in *CHI '14 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014, pp. 3181–3190.
- [5] K. Czuszyński, J. Ruminski, T. Kocejko, and J. Wtorek, "Septic safe interactions with smart glasses in health care," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015, pp. 1604–1607.
- [6] S. Ashley, "Touch-less control coming to cars," *S.A.E. J. Automot. Eng.*, vol. 1, no. 2, pp. 20–23, 2014.
- [7] F. Lardinois, "VW Debuts The Golf R Touch With All-Touchscreen User Interface, Gesture Control." [Online]. Available: <https://techcrunch.com/2015/01/05/vw-debuts-the-golf-r-touch-with-all-touchscreen-user-interface-gesture-control/>. [Accessed: 04-Mar-2018].
- [8] M. Stecher *et al.*, "Tracking Down the Intuitiveness of Gesture Interaction in the Truck Domain," *Procedia Manuf.*, vol. 3, no. Ahfe, pp. 3176–3183, 2015.
- [9] H. M. Mentis, "Voice or Gesture in the Operating Room," in *CHI EA '15 Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, 2015, pp. 773–780.
- [10] D. Gantenbein, "Kinect Launches a Surgical Revolution." [Online]. Available: <https://www.microsoft.com/en-us/research/blog/kinect-launches-surgical-revolution/>. [Accessed: 06-Mar-2018].
- [11] K. O'Hara *et al.*, "Touchless interaction in surgery," *Commun. ACM*, vol. 57, no. 1, pp. 70–77, 2014.
- [12] M. A. Belhaoua, M. A. Krebs, and M. J.-P. Radoux, "Gesture-Based Interaction on Surgical Field Using Touchless Technology," in *Reconnaissance des Formes et l'Intelligence Artificielle (RFIA'16)*, 2016, pp. 1–2.
- [13] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based Hand-gesture Applications," *Commun. ACM*, vol. 54, no. 2, 2011.
- [14] E. Nasr-Esfahani *et al.*, "Hand Gesture Recognition for Contactless Device Control in Operating Rooms," <https://arxiv.org/abs/1611.04138>, 2016.
- [15] N. Bizzotto, A. Costanzo, L. Bizzotto, D. Regis, A. Sandri, and B. Magnan, "Leap motion gesture control with OsiriX in the operating room to control imaging: First experiences during live surgery," *Surg. Innov.*, vol. 21, no. 6, pp. 655–656, 2014.
- [16] J. Nosta, "How Google Glass Is Changing Medical Education." [Online]. Available:

<https://www.forbes.com/sites/johnnosta/2013/06/27/google-glass-teach-me-medicine-how-glass-is-helping-change-medical-education/#177bc02429db>. [Accessed: 06-Mar-2018].

- [17] A. Mewes, B. Hensen, F. Wacker, and C. Hansen, "Touchless interaction with software in interventional radiology and surgery: a systematic literature review," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 12, no. 2, pp. 291–305, 2017.
- [18] "Antibiotic / Antimicrobial Resistance," *Centers for Disease Control and Prevention*. [Online]. Available: <https://www.cdc.gov/drugresistance/index.html>. [Accessed: 06-Mar-2018].
- [19] B. Huerta *et al.*, "Exploring the links between antibiotic occurrence, antibiotic resistance, and bacterial communities in water supply reservoirs," *Sci. Total Environ.*, vol. 456–457, pp. 161–170, 2013.
- [20] WHO, "Antibiotic resistance." [Online]. Available: <http://www.who.int/mediacentre/factsheets/antibiotic-resistance/en/>.
- [21] A. A. S. Sinegani and N. Younessi, "Antibiotic resistance of bacteria isolated from heavy metal-polluted soils with different land uses," *J. Glob. Antimicrob. Resist.*, vol. 10, pp. 247–255, 2017.
- [22] A. Kendon, "The Study of Gesture: Some Remarks on its History," in *Semiotics*, Springer, 1983, pp. 153–164.
- [23] M. Ishino and G. Stam, *Integrating gestures: The interdisciplinary nature of gesture*. John Benjamins Publishing Company, 2011.
- [24] R. M. Voyles and P. K. Khoslas, "Multi-Agent Gesture Interpretation for Robotic Cable Harnessing," in *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on, 1995*, pp. 1113–1118.
- [25] Y. Wu and T. S. Huang, "Human Hand Modeling , Analysis and Animation in the Context of," in *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on (Volume:3)*, 1999, pp. 6–10.
- [26] R. M. Krauss, R. a. Dushay, Y. Chen, and F. Rauscher, "The Communicative Value of Conversational Hand Gesture," *J. Exp. Soc. Psychol.*, vol. 31, no. 6, pp. 533–552, 1995.
- [27] M. Karam and M. C. Schraefel, "A Taxonomy of Gestures in Human Computer Interactions," *Tech. Report, Electron. Comput. Sci.*, pp. 1–45, 2005.
- [28] F. Quek, D. Mcneill, R. Bryll, and K. E. Mccullough, "Multimodal Human Discourse : Gesture and Speech University of Illinois at Chicago," *ACM Trans. Comput. Interact.*, vol. 9, no. 3, pp. 171–193, 2002.
- [29] Y. Wu and T. S. Huang, "Vision-based gesture recognition: A review," in *Gesture-Based Communication in Human-Computer Interaction*, 1999, pp. 1–12.
- [30] J. O. Wobbrock, M. R. Morris, and A. D. Wilson, "User-defined gestures for surface computing," in *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, 2009, pp. 1083–1092.
- [31] G. Murthy and R. Jadon, "A review of vision based hand gestures recognition," *Int. J. Inf. Technol. Knowl. Manag.*, vol. 2, no. 2, pp. 405–410, 2009.
- [32] H. Cheng, A. M. Chen, A. Razdan, and E. Buller, "Contactless Gesture Recognition for Mobile Devices," in *MIAA*, 2011, pp. 1–4.
- [33] Y. Yin, "Real-time continuous gesture recognition for natural multimodal interaction," Massachusetts Institute of Technology, 2014.
- [34] S. Berman and H. Stern, "Sensors for gesture recognition systems," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 42, no. 3, pp. 277–290, 2012.
- [35] Texas Instruments, "OPT3001 Ambient Light Sensor (ALS)," 2014. [Online]. Available:

- <http://www.ti.com/lit/ds/symlink/opt3001.pdf>. [Accessed: 05-Mar-2018].
- [36] Onsemi, “Ambient Light Sensor with I 2 C Interface and Dark Current Compensation,” 2012. [Online]. Available: http://www.onsemi.com/pub_link/Collateral/NOA1305-D.PDF. [Accessed: 05-Mar-2018].
- [37] Avago Technologies, “APDS-9009 Miniature Surface-Mount Ambient Light Photo Sensor.” [Online]. Available: <http://datasheet.octopart.com/APDS-9009-Avago-datasheet-62297739.pdf>. [Accessed: 05-Mar-2018].
- [38] A. Holmes, S. Desai, and A. Nahapetian, “LuxLeak: Capturing Computing Activity Using Smart Device Ambient Light Sensors,” in *Proceeding SmartObjects '16 Proceedings of the 2nd Workshop on Experiences in the Design and Implementation of Smart Objects*, 2016, pp. 47–52.
- [39] Sharp, “GP2Y0D21YK.” [Online]. Available: <https://www.sparkfun.com/datasheets/Components/GP2Y0A21YK.pdf>. [Accessed: 05-Mar-2018].
- [40] Silicon Labs, “Si1102,” 2010. [Online]. Available: <https://www.silabs.com/Support Documents/TechnicalDocs/Si1102.pdf>. [Accessed: 05-Mar-2018].
- [41] Vishay Semiconductors, “VCNL3020 Fully Integrated Proximity Sensor with Infrared Emitter, I2C Interface, and Interrupt Function.” [Online]. Available: <http://www.vishay.com/docs/84150/vcnl3020.pdf>. [Accessed: 05-Mar-2018].
- [42] Avago Technologies, “HSDL-9100 Surface-Mount Proximity Sensor.” [Online]. Available: <https://www.mouser.com/ds/2/678/V02-2259EN0-1100110.pdf>. [Accessed: 05-Mar-2018].
- [43] K. Hinckley, J. Pierce, M. Sinclair, and E. Horvitz, “Sensing Techniques for Mobile Interaction,” in *UIST '00 Proceedings of the 13th annual ACM symposium on User interface software and technology*, 2000, pp. 91–100.
- [44] C. Metzger, M. Anderson, and T. Starner, “FreeDigiter: A Contact – free Device for Gesture Control,” in *ISWC '04 Proceedings of the Eighth International Symposium on Wearable Computers*, 2004, pp. 18–21.
- [45] H. Manabe, “Multi-touch gesture recognition by single photoreflector,” in *UIST '13 Adjunct Proceedings of the adjunct publication of the 26th annual ACM symposium on User interface software and technology*, 2013, pp. 15–16.
- [46] Y. Gao, A. M. Broga, and P. Krishnaswamy, “Contactless gesture recognition with sensor having asymmetric field of view,” EP2866124 A1, 2015.
- [47] K. Czuszynski, J. Ruminski, J. Wtorek, A. Vogl, and M. Haller, “Interactions using passive optical proximity detector,” in *Human System Interactions (HSI), 2015 8th International Conference on*, 2015, pp. 180–186.
- [48] H. Cheng, A. M. Chen, A. Razdan, and E. Buller, “Contactless Gesture Recognition System Using Proximity Sensors,” in *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, 2011, no. 1, pp. 1–2.
- [49] A. Chen, H.-T. Cheng, A. Razdan, and E. Buller, “Methods and apparatus for contactless gesture recognition,” U.S. Patent App. 13/161,955, 2011.
- [50] T. Chang, K. P. Wu, C. J. Fang, C. Te Chan, C. T. Chuang, and F. Y. Liu, “Light sensor system for object detection and gesture recognition, and object detection method,” U.S. Patent App. 13/494,000, 2012.
- [51] Y. S. Kim and K. Baek, “A motion gesture sensor using photodiodes with limited field-of-view,” *Opt. Express*, vol. 21, no. 8, pp. 555–560, 2013.
- [52] K. Kong, Y. S. Kim, J. E. Kim, S. Kim, and K. Baek, “Single-Package Motion Gesture Sensor

- for Portable Applications,” *IEEE Trans. Consum. Electron.*, vol. 59, no. 4, pp. 848–853, 2013.
- [53] J. S. Kim, S. J. Yun, D. J. Seol, H. J. Park, and Y. S. Kim, “An IR Proximity-Based 3D Motion Gesture Sensor for Low-Power Portable Applications,” *IEEE Sens. J.*, vol. 15, no. 12, pp. 7009–7016, 2015.
- [54] J. S. Kim, S. J. Yun, and Y. S. Kim, “Low-power motion gesture sensor with a partially open cavity package,” *Opt. Express*, vol. 24, no. 10, pp. 10537–10546, 2016.
- [55] Z. Zivkovic, “Air Gesture Control Using 5-Pixel Light Sensor,” in *Consumer Electronics (ICCE), 2014 IEEE International Conference on*, 2014, pp. 1–2.
- [56] A. D. Wilson, “TouchLight - An Imaging Touch Screen and Display for Gesture-Based Interaction,” *Proc. 6th Int. Conf. Multimodal Interfaces*, pp. 69–76, 2004.
- [57] C. Chuang, T. Chang, P. Jau, and F. Chang, “Touchless Positioning System Using LED Sensors,” in *System Science and Engineering (ICSSE), 2014 IEEE International Conference on*, 2014, pp. 1–6.
- [58] C. Chuang, T. Chang, P. Jau, and F. Chang, “Applying the Kalman Filter to the Infrared-Based Touchless Positioning System with Dynamic Adjustment of Measurement Noise Features,” in *Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT), 2014 9th International*, 2014, pp. 84–87.
- [59] A. Butler, S. Izadi, and S. Hodges, “SideSight: multi-‘touch’ interaction around small devices,” in *UIST '08 Proceedings of the 21st annual ACM symposium on User interface software and technology*, 2008, pp. 201–204.
- [60] S. K. Tang, W. C. Tseng, W. W. Luo, K. C. Chiu, S. T. Lin, and Y. P. Liu, “Virtual Mouse: A Low Cost Proximity-Based Gestural Pointing Device,” in *Human-Computer Interaction. Interaction Techniques and Environments: 14th International Conference, HCI International 2011, Orlando, FL, USA, July 9-14, 2011, Proceedings, Part II*, 2011, pp. 491–499.
- [61] Á. Tar and G. Csereny, “Object outline and surface-trace detection using infrared proximity array,” *IEEE Sens. J.*, vol. 11, no. 10, pp. 2486–2493, 2011.
- [62] V. Pavlov, H. Ruser, and M. Horn, “Feature extraction from an infrared sensor array for localization and surface recognition of moving cylindrical objects,” *2007 IEEE Instrum. Meas. Technol. Conf. IMTC 2007*, pp. 1–6, 2007.
- [63] G. Batchuluun, B. Odgerel, and C. H. Lee, “Hand Gesture Recognition Using an Infrared Proximity Sensor Array,” *Int. J. Fuzzy Log. Intell. Syst.*, vol. 15, no. 3, pp. 186–191, 2015.
- [64] S. R. Das, T. E. Jakulin, and K. G. J. Nigel, “Linear and rotational air gesture detection using optical sensors setup in automotive infotainment system,” *Int. Conf. Commun. Signal Process. ICCSP 2016*, pp. 1164–1169, 2016.
- [65] A. Withana, R. Peiris, N. Samarasekara, and S. Nanayakkara, “zSense: Enabling Shallow Depth Gesture Recognition for Greater Input Expressivity on Smart Wearables,” in *CHI '15 Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 3661–3670.
- [66] A. Withana *et al.*, “waveSense: Ultra Low Power Gesture Sensing Based on Selective Volumetric Illumination,” in *UIST '16 Adjunct Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016, pp. 139–140.
- [67] C. Zhang, J. Tabor, J. Zhang, and X. Zhang, “Extending Mobile Interaction Through Near-Field Visible Light Sensing,” in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking - MobiCom '15*, 2015, pp. 345–357.
- [68] Sharp, “GP2AP054A00F Proximity/Gesture Sensor with Integrated Ambient Light Sensor,” 2002. [Online]. Available: <http://www.sharp->

world.com/products/device/lineup/data/pdf/datasheet/gp2ap054a00f_e.pdf. [Accessed: 05-Mar-2018].

- [69] Avago Technologies, “APDS-9960 Digital Proximity, Ambient Light, RGB and Gesture Sensor.” [Online]. Available: https://cdn.sparkfun.com/assets/learn_tutorials/3/2/1/Avago-APDS-9960-datasheet.pdf. [Accessed: 04-Mar-2018].
- [70] Analog Devices, “Photometric Sensor for Gesture and Proximity ADUX1020 * Product Page Quick Links,” 2016. [Online]. Available: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADUX1020.pdf>. [Accessed: 12-Dec-2017].
- [71] J. Kim, J. He, K. Lyons, and T. Starner, “The Gesture Watch: A wireless contact-free Gesture based wrist interface,” *Proc. - Int. Symp. Wearable Comput. ISWC*, pp. 15–22, 2007.
- [72] S. Kratz and M. Rohs, “HoverFlow: Expanding the Design Space of Around-device Interaction,” in *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2009, p. 4:1–4:8.
- [73] M. Kaholokula, “Reusing Ambient Light to Recognize Hand Gestures,” <http://www.cs.dartmouth.edu/reports/TR2016-797.pdf>, pp. 1–17, 2016.
- [74] P. Wojtczuk, A. Armitage, T. D. Binnie, and T. Chamberlain, “PIR Sensor Array for Hand Motion Recognition,” in *SENSORDEVICES 2011 : The Second International Conference on Sensor Device Technologies and Applications*, 2011, no. c, pp. 99–102.
- [75] P. Wojtczuk, D. Binnie, A. Armitage, T. Chamberlain, and C. Giebler, “A Touchless Passive Infrared Gesture Sensor,” in *UIST '13 Adjunct Proceedings of the adjunct publication of the 26th annual ACM symposium on User interface software and technology*, 2013, pp. 67–68.
- [76] I. Agarwal and S. Johar, “A Tutor For The Hearing Impaired (Developed Using Automatic Gesture Recognition),” *Int. J. Comput. Sci. Eng. Appl.*, vol. 1, no. 4, pp. 49–61, 2011.
- [77] T. Kadir, R. Bowden, E. J. Ong, and A. Zisserman, “Minimal Training, Large Lexicon, Unconstrained Sign Language Recognition,” *Proceedings Br. Mach. Vis. Conf. 2004*, p. 96.1-96.10, 2004.
- [78] Y. Araga, M. Shirabayashi, K. Kaida, and H. Hikawa, “Real time gesture recognition system using posture classifier and Jordan recurrent neural network,” in *Proceedings of the International Joint Conference on Neural Networks*, 2012, pp. 10–15.
- [79] M. Hasanuzzaman, V. Ampornaramveth, M. a. Bhuiyan, Y. Shirai, and H. Ueno, “Real-time Vision-based Gesture Recognition for Human Robot Interaction,” *2004 IEEE Int. Conf. Robot. Biomimetics*, pp. 413–418, 2004.
- [80] S. Marcel, O. Bernier, J. E. Viallet, and D. Collobert, “Hand gesture recognition using input-output hidden Markov models,” in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, 2000, pp. 456–461.
- [81] A. Pasarica *et al.*, “Remote Control of a Robotic Platform Based on Hand Gesture Recognition,” in *E-Health and Bioengineering Conference (EHB), 2017*, 2017, no. 2006, pp. 643–646.
- [82] A. Maqueda, C. Del-Blanco, F. Jaureguizar, and N. García, “Human – computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns,” *Comput. Vis. Image Underst.*, vol. 141, pp. 126–137, 2015.
- [83] E. Ueda, Y. Matsumoto, M. Imai, and T. Ogasawara, “A Hand-Pose Estimation for Vision-Based Human Interfaces,” *IEEE Trans. Ind. Electron.*, vol. 50, no. 4, pp. 676–684, 2003.
- [84] J. Rekimoto, “Brightshadow: shadow sensing with synchronous illuminations for robust gesture recognition,” *Proc. ACM CHI 2008 Conf. Hum. Factors Comput. Syst.*, vol. 2, pp. 2769–2774, 2008.

- [85] Y. Chen, Z. Ding, Y. L. Chen, and X. Wu, "Rapid recognition of dynamic hand gestures using leap motion," in *2015 IEEE International Conference on Information and Automation, ICIA 2015 - In conjunction with 2015 IEEE International Conference on Automation and Logistics*, 2015, pp. 1419–1424.
- [86] C.-H. Chuan, E. Regina, and C. Guardino, "American Sign Language Recognition Using Leap Motion Sensor," in *2014 13th International Conference on Machine Learning and Applications*, 2014, pp. 541–544.
- [87] L. Ma and W. Huang, "A static hand gesture recognition method based on the depth information," in *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2016 8th International Conference on*, 2016, pp. 136–139.
- [88] T. Liu, W. Zhou, and H. Li, "Sign language recognition with long short-term memory," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 2871–2875.
- [89] K. Stefanov and J. Beskow, "Gesture Recognition System for Isolated Sign Language Signs," in *The 4th European and 7th Nordic Symposium on Multimodal Communication, 29-30 September 2016, University of Copenhagen, Denmark*, 2016, pp. 57–59.
- [90] M. Tornow, A. Al-hamadi, and V. Borrmann, "Gestic-Based Human Machine Interface for Robot Control," in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, 2013, pp. 2706–2711.
- [91] S. Gupta, P. Molchanov, X. Yang, K. Kim, S. Tyree, and J. Kautz, "Towards Selecting Robust Hand Gestures for Automotive Interfaces," in *Intelligent Vehicles Symposium (IV), 2016 IEEE*, 2016, pp. 1350–1357.
- [92] D. Ryu, D. Um, P. Tanofsky, D. H. Koh, Y. S. Ryu, and S. Kang, "T-less: A novel touchless human-machine interface based on infrared proximity sensing," *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, pp. 5220–5225, 2010.
- [93] Z. Lv, A. Halawani, M. Sikandar, and L. Khan, "Finger in Air: Touch-less Interaction on Smartphone," in *12th International Conference on Mobile and Ubiquitous Multimedia (MUM 2013)*, 2013, pp. 1–4.
- [94] H. Lahiani and M. Elleuch, "Real Time Hand Gesture Recognition System for Android Devices," in *15th International Conference on Intelligent Systems Design and Applications (ISDA)*, 2015, pp. 591–596.
- [95] W. Wong, S. Hsu, and C. Huang, "Virtual TouchPad: Hand Gesture Recognition For Smartphone With Depth Camera," in *Consumer Electronics - Taiwan (ICCE-TW), 2015 IEEE International Conference on*, 2015, no. 4, pp. 214–215.
- [96] F. Aezinia, Y. Wang, and B. Bahreyni, "Touchless Capacitive Sensor for Hand Gesture Detection," in *SENSORS, 2011 IEEE*, 2011, pp. 3–6.
- [97] M. Pulkkinen, J. Salomaa, M. M. Moayer, T. Haapala, and K. Halonen, "462-nW 2-Axis Gesture Sensor Interface Based On Capacitively Controlled Ring Oscillators," in *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*, 2017, pp. 7–10.
- [98] B. Kellogg, V. Talla, and S. Gollakota, "Bringing Gesture Recognition To All Devices," in *Proceeding NSDI'14 Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, 2014, pp. 303–316.
- [99] J. Lien *et al.*, "Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar," *ACM Trans. Graph. Artic.*, vol. 35, no. 10, pp. 1–19, 2016.
- [100] S. Y. Kim, H. G. Han, J. W. Kim, S. Lee, and T. W. Kim, "A Hand Gesture Recognition Sensor Using Reflected Impulses," *IEEE Sens. J.*, vol. 17, no. 10, pp. 2975–2976, 2017.
- [101] T. Fan *et al.*, "Wireless Hand Gesture Recognition Based on Continuous-Wave Doppler Radar

- Sensors,” *IEEE Trans. Microw. Theory Tech.*, vol. 64, no. 11, pp. 4012–4020, 2016.
- [102] T. Fan, D. Ye, J. Hangfu, Y. Sun, C. Li, and L. Ran, “Hand Gesture Recognition Based on Wi-Fi Chipsets,” in *Radio and Wireless Symposium (RWS), 2017 IEEE*, 2017, pp. 98–100.
- [103] M. Tang, F. Wang, and T. Horng, “Human Gesture Sensor Using Ambient Wireless Signals Based on Passive Radar Technology,” in *Microwave Symposium (IMS), 2015 IEEE MTT-S International*, 2015, no. 1, pp. 2–5.
- [104] F. Wang, M. Tang, Y. Chiu, and T. Horng, “Gesture Sensing Using Retransmitted Wireless Communication Signals Based on Doppler Radar Technology,” *IEEE Trans. Microw. Theory Tech.*, vol. 63, no. 12, pp. 4592–4602, 2015.
- [105] R. Krigslund, P. Popovski, and G. F. Pedersen, “3D Gesture Recognition Using Passive RFID Tags,” in *Antennas and Propagation Society International Symposium (APSURSI), 2013 IEEE*, 2013, pp. 2307–2308.
- [106] B. Van Dam, Y. Murillo, M. Li, and S. Pollin, “In-air ultrasonic 3D-touchscreen with gesture recognition using existing hardware for smart devices,” *IEEE Work. Signal Process. Syst. SiPS Des. Implement.*, pp. 74–79, 2016.
- [107] A. Bujnowski *et al.*, “Comparison of active proximity radars for the wearable devices,” in *Proceedings - 2015 8th International Conference on Human System Interaction, HSI 2015*, 2015, pp. 158–165.
- [108] V. I. Pavlovic, R. Sharma, and T. S. Huang, “Visual Interpretation of Hand Gestures for Human-Computer Interaction: a Review,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 677–695, 1997.
- [109] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning - Systemy uczące się*, I. PWN, 2018.
- [110] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio, “An empirical evaluation of deep architectures on problems with many factors of variation,” *Proc. 24th Int. Conf. Mach. Learn. - ICML '07*, no. 2006, pp. 473–480, 2007.
- [111] J. Bergstra and Y. Bengio, “Random Search for Hyper-Parameter Optimization,” *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.
- [112] C. Carley and C. Tomasi, “Single-Frame Indexing for 3D Hand Pose Estimation,” in *Computer Vision Workshop (ICCVW), 2015 IEEE International Conference on*, 2015, pp. 493–501.
- [113] H. Stern, K. Smilansky, and S. Berman, “Depth Based Dual Component Dynamic Gesture Recognition,” in *IPCV'13 - The 2013 International Conference on Image Processing and Computer Vision*, 2013.
- [114] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, “uWave: Accelerometer-based personalized gesture recognition and its applications,” *Pervasive Mob. Comput.*, vol. 5, no. 6, pp. 657–675, 2009.
- [115] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
- [116] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.
- [117] S. Wu, S. Zhong, and Y. Liu, “Deep residual learning for image steganalysis,” *Multimed. Tools Appl.*, pp. 1–17, 2017.
- [118] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [119] M. Ariyanto *et al.*, “Finger Movement Pattern Recognition Method Using Artificial Neural Network Based on Electromyography (EMG) Sensor,” in *Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT), 2015*

International Conference on, 2015, pp. 12–17.

- [120] P. R. Futane and R. V. Dharaskar, “General Fuzzy Minmax Neural Network For Subset Of Indian Sign Language,” in *Hybrid Intelligent Systems (HIS), 2012 12th International Conference on*, 2012, pp. 525–530.
- [121] M. Yang, N. Ahuja, and M. Tabb, “Extraction of 2D Motion Trajectories and Its Application to Hand Gesture Recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1061–1074, 2002.
- [122] E. Tsironi, P. Barros, C. Weber, and S. Wermter, “An analysis of Convolutional Long Short-Term Memory Recurrent Neural Networks for gesture recognition,” *Neurocomputing*, vol. 268, pp. 76–86, 2017.
- [123] X. Chai, Z. Liu, F. Yin, Z. Liu, and X. Chen, “Two Streams Recurrent Neural Networks for Large-Scale Continuous Gesture Recognition,” in *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2016.
- [124] P. Wang, W. Li, C. Li, and Y. Hou, “Action Recognition Based on Joint Trajectory Maps with Convolutional Neural Networks,” <https://arxiv.org/abs/1612.09401>, 2016.
- [125] K.-T. Cheng and A. S. Krishnakumar, “Automatic Functional Test Generation Using The Extended Finite State Machine Model,” in *Design Automation Conference - DAC*, 1993, pp. 86–91.
- [126] V. I. Ulyantsev and F. N. Tsarev, “Extended Finite-State Machine Induction using SAT-Solver,” in *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, 2011, pp. 512–517.
- [127] H. Kopetz, B. Huber, and R. Obermaisser, “Periodic Finite-State Machines,” in *Object and Component-Oriented Real-Time Distributed Computing, 2007. ISORC '07. 10th IEEE International Symposium on*, 2007.
- [128] P. Hong, M. Turk, and T. S. Huang, “Constructing Finite State Machines for Fast Gesture Recognition,” in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, 2000, pp. 691–694.
- [129] T. E. Starner and A. Pentland, “Visual Recognition of American Sign Language Using Hidden Markov Models,” *Massachusetts Inst Of Tech Cambridge Dept Of Brain And Cognitive Sciences*, pp. 189–194, 1995.
- [130] C. W. Ng and S. Ranganath, “Gesture recognition via pose classification,” in *Proceedings - International Conference on Pattern Recognition*, 2000, vol. 15, no. 3, pp. 699–704.
- [131] P. Premaratne, S. Yang, P. Vial, and Z. Iftikhar, “Centroid tracking based dynamic hand gesture recognition using discrete Hidden Markov Models,” *Neurocomputing*, vol. 228, no. October 2016, pp. 79–83, 2017.
- [132] J. Elman, “Finding structure in time,” *Cogn. Sci.*, vol. 14, no. 1 990, pp. 179–211, 1990.
- [133] A. Géron, *Hands-On Machine Learning with Scikit-Learn & TensorFlow*. O’Reilly Media Inc., 2017.
- [134] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training Recurrent Neural Networks,” in *Proceeding ICML’13 Proceedings of the 30th International Conference on International Conference on Machine Learning*, 2013, pp. 1310–1318.
- [135] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [136] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” <https://arxiv.org/abs/1412.3555>, pp. 1–9, 2014.
- [137] S. Oprea, A. Garcia-Garcia, J. Garcia-Rodriguez, S. Orts-Escolano, and M. Cazorla, “A

- recurrent neural network based Schaeffer gesture recognition system,” in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 425–431.
- [138] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, “DiSAN: Directional Self-Attention Network for RNN/CNN-free Language Understanding,” <https://arxiv.org/abs/1709.04696>, 2017.
- [139] W. Hinoshita, T. Ogata, H. Kozima, H. Kanda, T. Takahashi, and H. G. Okuno, “Emergence of evolutionary interaction with voice and motion between two robots using RNN,” *2009 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS 2009*, pp. 4186–4192, 2009.
- [140] D. Bukhari, Y. Wang, and H. Wang, “Multilingual Convolutional, Long Short-Term Memory, Deep Neural Networks for Low Resource Speech Recognition,” *Procedia Comput. Sci.*, vol. 107, pp. 842–847, 2017.
- [141] N. Neverova, C. Wolf, G. Paci, G. Somnavilla, G. W. Taylor, and F. Nebout, “A multi-scale approach to gesture detection and recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 484–491.
- [142] S. Shin and W. Sung, “Dynamic hand gesture recognition for wearable devices with low complexity recurrent neural networks,” in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016, pp. 2274–2277.
- [143] A. van Dam, “Beyond WIMP,” *IEEE Comput. Graph. Appl.*, vol. 20, no. 1, pp. 50–51, 2000.
- [144] M. Teruel, “Improving collaborative and Post-WIMP systems through requirements specification,” in *Requirements Engineering Conference (RE), 2014 IEEE 22nd International*, 2014.
- [145] T. Piumsomboon, A. Clark, M. Billingham, and A. Canterbury, “User-Defined Gestures for Augmented Reality,” in *CHI EA '13 CHI '13 Extended Abstracts on Human Factors in Computing Systems*, 2013, pp. 282–299.
- [146] E. Chan, T. Seyed, W. Stuerzlinger, X.-D. Yang, and F. Maurer, “User Elicitation on Single-hand Microgestures,” in *CHI '16 Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 3403–3414.
- [147] L. C. Henyey and J. L. Greenstein, “Diffuse radiation in the galaxy,” *Astrophys. J.*, vol. 93, pp. 70–83, 1941.
- [148] A. Krishnaswamy and G. V. G. Baranoski, “A biophysically-based spectral model of light interaction with human skin,” *Comput. Graph. Forum*, vol. 23, no. 3 SPEC. ISS., pp. 331–340, 2004.
- [149] G. Hall, S. L. Jacques, K. W. Eliceiri, and P. J. Campagnola, “Goniometric measurements of thick tissue using Monte Carlo simulations to obtain the single scattering anisotropy coefficient,” *Biomed. Opt. Express*, vol. 3, no. 11, p. 2707, 2012.
- [150] C. Zhu and Q. Liu, “Review of Monte Carlo modeling of light transport in tissues Review of Monte Carlo modeling of light transport,” *J. Biomed. Opt.*, vol. 18, no. 5, pp. 1–12, 2013.
- [151] T. F. Chen, G. V. G. Baranoski, B. W. Kimmel, and E. Miranda, “Hyperspectral Modeling of Skin Appearance,” *ACM Trans. Graph.*, vol. 34, no. 3, pp. 1–14, 2015.
- [152] P. Hanrahan and W. Krueger, “Reflection from Layered Surfaces Due to Subsurface Scattering,” *Proc. 20th Annu. Conf. Comput. Graph. Interact. Tech.*, pp. 165–174, 1993.
- [153] J. H. Lee, P. K. J. Park, C. Shin, H. Ryu, B. C. Kang, and T. Delbruck, “Touchless hand gesture UI with instantaneous responses,” in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 1957–1960.
- [154] K. Czuszynski, J. Ruminski, A. Polinski, and A. Bujnowski, “Estimation of the amplitude of the signal for the active optical gesture sensor with sparse detectors,” in *Human System*

Interactions (HSI), 2016 9th International Conference on, 2016, pp. 483–489.

- [155] J. Han, L. Shao, D. Xu, and J. Shotton, “Enhanced Computer Vision with Microsoft Kinect Sensor: A Review,” *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, 2013.
- [156] K. Czuszyński, J. Ruminski, and J. Wtorek, “Pose classification in the gesture recognition using the linear optical sensor,” in *Human System Interactions (HSI), 2017 10th International Conference on*, 2017, pp. 18–24.
- [157] K. Czuszyński, J. Ruminski, and J. Wtorek, “Analysis of the properties of the active linear gesture sensor,” *Metrol. Meas. Syst.*, vol. 24, no. 4, pp. 617–630, 2017.
- [158] K. Czuszyński, J. Ruminski, and J. Wtorek, “The passive operating mode of the linear optical gesture sensor,” *Adv. Electr. Comput. Eng.*, vol. 18, no. 1, pp. 145–156, 2018.
- [159] R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.
- [160] R. Bainbridge and J. A. Paradiso, “Wireless Hand Gesture Capture Through Wearable Passive Tag Sensing,” in *Body Sensor Networks (BSN), 2011 International Conference on*, 2011, pp. 200–204.
- [161] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2012.
- [162] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, “Self-Normalizing Neural Networks,” <https://arxiv.org/abs/1706.02515>, 2017.
- [163] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *International Conference on Learning Representations (ICLR)*, 2015, pp. 1–15.
- [164] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima,” in *International Conference on Learning Representations (ICLR)*, 2017, pp. 1–16.
- [165] A. Karpathy and L. Fei-Fei, “Deep Visual-Semantic Alignments for Generating Image Descriptions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 664–676, 2017.
- [166] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, “Hyperopt: A Python library for model selection and hyperparameter optimization,” *Comput. Sci. Discov.*, vol. 8, 2015.
- [167] K. Czuszyński, J. Ruminski, A. Bujnowski, and J. Wtorek, “Semi complex navigation with an active optical gesture sensor,” in *UbiComp '16 Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, 2016, pp. 269–272.
- [168] “Department of Defense Design Criteria Standard - Human Engineering,” 1999.
- [169] K. Czuszyński, J. Ruminski, and A. Kwasniewska, “Gesture Recognition with the Linear Optical Sensor and Recurrent Neural Networks,” *IEEE Sens. J.*, 2018.

DODATEK A: DOROBEK NAUKOWY MGR. INŻ. KRZYSZTOFA CZUSZYŃSKIEGO

Publikacje bezpośrednio związane z tematyką rozprawy doktorskiej

Z listy JCR:

1. **K. Czuszyński**, J. Rumiński, J. Wtorek, "Analysis of the properties of the active linear gesture sensor", *Metrology and Measurement Systems*, vol. 24, issue 4, 2017, strony: 617-630,
2. **K. Czuszyński**, J. Rumiński, J. Wtorek, "The passive operating mode of the linear optical gesture sensor", *Advances in Electrical and Computer Engineering*, vol. 18, issue 1, 2018, strony: 145-156.

Konferencyjne, indeksowane w Web of Science:

3. **K. Czuszyński**, J. Rumiński, J. Wtorek, A. Vogl, M. Haller, "Interactions using passive optical proximity detector", *Human System Interactions (HSI), 2015 8th International Conference on*, 2015, strony: 180-186,
4. A. Bujnowski, **K. Czuszyński**, J. Rumiński, J. Wtorek, R. McCall, A. Popleteev, N. Louveton, T. Engel, "Comparison of active proximity radars for the wearable devices", *Human System Interactions (HSI), 2015 8th International Conference on*, 2015, strony: 158-165,
5. **K. Czuszyński**, J. Rumiński, T. Kocejko, J. Wtorek, "Septic safe interactions with smart glasses in health care", *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015, strony: 1604-1607,
6. **K. Czuszyński**, J. Rumiński, A. Bujnowski, J. Wtorek, "Semi complex navigation with an active optical gesture sensor", *UbiComp '16 Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, 2016, strony: 269-277,

7. **K. Czuszyński**, J. Rumiński, A. Poliński, A. Bujnowski, "Estimation of the amplitude of the signal for the active optical gesture sensor with sparse detectors", *Human System Interactions (HSI), 2016 9th International Conference on*, 2016, strony: 483-489, **Best Paper Award**,
8. **K. Czuszyński**, J. Rumiński, J. Wtorek, "Pose classification in the gesture recognition using the linear optical sensor", *Human System Interactions (HSI), 2017 10th International Conference on*, 2017, strony: 18-24, **Best Paper Award**.

Zaakceptowane do publikacji w czasopismach z listy JCR:

9. **K. Czuszyński**, J. Rumiński, A. Kwaśniewska, "Gesture Recognition with the Linear Optical Sensor and Recurrent Neural Networks", *IEEE Sensors*, 2018.

Pozostałe publikacje

Z listy JCR:

10. P. Winklewski, Y. Tkachenko, K. Mazur, J. Kot, M. Gruszecki, W. Gumiński, **K. Czuszyński**, J. Wtorek, A. Frydrychowski, "Sympathetic Activation Does Not Affect the Cardiac and Respiratory Contribution to the Relationship between Blood Pressure and Pial Artery Pulsation Oscillations in Healthy Subjects", *PloS one*, vol. 10, issue 8, 2015, strony: 1-13,
11. M. Biallas, A. Andrushevich, R. Kistler, A. Klapproth, **K. Czuszyński**, A. Bujnowski, "Feasibility Study for Food Intake Tasks Recognition Based on Smart Glasses", *Journal of Medical Imaging and Health Informatics*, vol. 5, issue 8, 2015, strony: 1688-1694.

Konferencyjne, indeksowane w Web of Science:

12. **K. Czuszyński**, J. Rumiński, "Interaction with medical data using QR-codes", *Human System Interactions (HSI), 2014 7th International Conference on*, 2014, strony: 182-187,
13. J. Rumiński, **K. Czuszyński**, "Application of smart glasses for fast and automatic color correction in health care", 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015, strony: 4950-4953,

14. A. Bujnowski, J. Rumiński, M. Kaczmarek, **K. Czuszyński**, P. Przystup, "Cardiovascular data analysis using electronic wearable eyeglasses-preliminary study", *Computer Science and Information Systems (FedCSIS)*, 2016 Federated Conference on, 2016, strony: 1409-1412,
15. J. Rumiński, A. Bujnowski, **K. Czuszyński**, T. Kocejko, "Estimation of respiration rate using an accelerometer and thermal camera in eGlasses", *Computer Science and Information Systems (FedCSIS)*, 2016 Federated Conference on, 2016, strony: 1431-1434,
16. T. Kocejko, A. Bujnowski, J. Rumiński, **K. Czuszyński**, M. Pietrewicz, "EMG and gaze based interaction with graphic interface of smart glasses application", *Human System Interactions (HSI)*, 2016 9th International Conference on, 2016, strony: 476-482,
17. T. Kocejko, **K. Czuszyński**, J. Rumiński, A. Bujnowski, A. Poliński, J. Wtorek, "Extending touch-less interaction with smart glasses by implementing EMG module", *Human System Interactions (HSI)*, 2017 10th International Conference on, 2017, strony: 12-17.

Pozostale:

18. **K. Czuszyński**, J. Wtorek, "Blood pressure estimation by means of a joint impedance – photoplethysmographic method", *ICT Young Conference 2013*, strony: 81-86, **Best Paper Award**,
19. **K. Czuszyński**, J. Wtorek, "Bezprzewodowy moduł detekcji źródeł dźwięku – system klasyfikatorów", *Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne*, 2013, strony: 1205-1211,
20. **K. Czuszyński**, "Noninvasive evaluation of cardiac output based on various biosignals", *PhD Interdisciplinary Journal*, vol. 1, strony: 115-124,
21. **K. Czuszyński**, A. Gorczewska, "Impact of surface skin temperature change on blood flow characteristics in palm", *ICT Young Conference 2014*, strony: 47-52, **Best Paper Award**.

Udział w projekcie badawczym

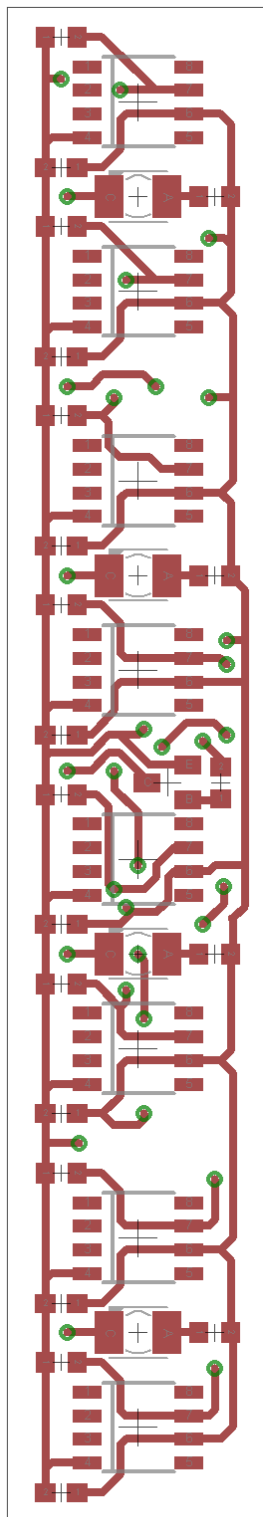
Wykonawca w projekcie *Interaktywne okulary dla mobilnego i percepcyjnego przetwarzania (eGlasses)* w latach 2014-2016.

Wskaźnik cytowań

Autor niniejszej rozprawy posiada w swoim dorobku (stan na 21 marca 2018 roku) z opublikowanych pozycji w sumie 4 publikacje w czasopismach z listy JCR oraz 16 publikacji konferencyjnych lub pozostałych, które były cytowane łącznie 51 razy według Web of Science (Core Collection) i 79 razy według Google Scholar. Indeks Hirsha autora wynosi 4 według Web of Science (Core Collection) i 5 według Google Scholar.

DODATEK B SCHEMAT PCB

Strona górna



Strona dolna

