Politechnika Śląska
Wydział Automatyki, Elektroniki i Informatyki

Mgr inż. Tomasz Grzejszczak

WYKRYWANIE I LOKALIZACJA PUNKTÓW CHARAKTERYSTYCZNYCH DŁONI W OBRAZACH CYFROWYCH

Rozprawa doktorska napisana pod kierunkiem dr. hab. inż. Adama Gałuszki

Promotor pomocniczy:
Dr inż. Michał Kawulok

Gliwice 2014

Niektóre badania prowadzone w ramach niniejszej rozprawy doktorskiej były współfinansowane z następujących projektów:

"Fundusz Stypendiów dla doktorantów na rzecz rozwoju transferu wiedzy w regionie" realizowany w ramach Programu Operacyjnego Kapitał Ludzki, Poddziałanie 8.2.1 Wsparcie dla współpracy sfery nauki i przedsiębiorstw. Projekt realizowany przez Technopark Gliwice Sp. z o.o.

Projekt współfinansowany *ze* środków Unii Europejskiej *w* ramach Europejskiego Funduszu Społecznego

Projekt badawczy nr IP2011 023071: „Detekcja dłoni oraz rozpoznawanie jej ułożenia na potrzeby kreowania komunikacji wizyjnej pomiędzy człowiekiem a komputerem" realizowanym w ramach programu „Iuventus Plus" (M. P. Nr 61, poz. 587).

„DoktoRIS - Program stypendialny na rzecz innowacyjnego Śląska" realizowany w ramach Priorytetu VIII Programu Operacyjnego Kapitał Ludzki 2007-2013 Regionalne Kadry Gospodarki, Działania 8.2 Transfer wiedzy, Poddziałania 8.2.2 Regionalne Strategie Innowacji.

Projekt współfinansowany *ze* środków Unii Europejskiej *w* ramach Europejskiego Funduszu Społecznego

Projekty badawcze BKM finansowane ze środków Instytutu Automatyki przeznaczone na badania kierunkowe

Spis treści

Spis skrótów 5
Spis symboli i oznaczeń 5
1. Wstęp 6
1.1. Uzasadnienie konieczności badań 7
1.2. Układ pracy 7
1.3. Zakres pracy 8
1.4. Cel badań 11
1.5. Ułożenie i stopnie swobody dłoni 12
1.6. Teza pracy 14
2. Analiza stanu wiedzy 15
2.1. Przegląd literatury 16
2.1.1. Systemy sensoryczne, niewizyjne oraz systemy wizyjne wspomagane 16
2.1.2. Systemy wizyjne oparte o kamery inne niż światła widzialnego 17
2.1.3. Systemy wizyjne oparte na kamerach światła widzialnego 19
2.2. Założenia dotyczące zakresu badań 20
2.3. Przegląd istniejących metod wykrywania punktów charakterystycznych 21
2.3.1. Analiza odległości punktów konturu od nadgarstka 22
2.3.2. Analiza transformaty odległościowej 24
2.3.3. Dopasowanie wzorca koła do maski dłoni 26
2.3.4. Samoorganizująca się mapa neuronowa 28
3. Wybrane metody sztucznej inteligencji i przetwarzania obrazów 33
3.1. Metody sztucznej inteligencji 33
3.1.1. Wnioskowanie 33
3.1.2. Drzewo decyzyjne 34
3.2. Metody analizy obrazu 35
3.2.1. Metody segmentacji obszaru występowania skóry 35
3.2.2. Transformacja odległościowa 36

3.2.3. Dopasowanie wzorców 37
3.2.4. Obraz kierunkowy 39
4. Propozycja autorskiego algorytmu wykrywania punktów charakterystycznych 42
4.1. Wykrywanie i lokalizacja punktów nadgarstka 42
4.1.1. Propozycje ulepszenia metody wykrywania i lokalizacji punktów nadgarstka 47
4.2. Wykorzystanie wybranych metod analizy obrazu do lokalizacji punktów charakterystycznych 48
4.3. Selekcja i klasyfikacja wykrytych punktów charakterystycznych 54
5. Walidacja eksperymentalna 62
5.1. Zbiór gestów i baza obrazów 62
5.2. Procedura walidacyjna 65
5.2.1. Ocena metod wykrywania nadgarstka 65
5.2.2. Ocena metod wykrywania pozostałych punktów charakterystycznych 66
5.2.3. Metoda wyznaczania parametrów sterujących 71
5.3. Analiza wyników 71
5.3.1. Badanie metod poszukiwania punktów nadgarstka 71
5.3.2. Badanie metod wykrywania pozostałych punktów charakterystycznych 75
5.3.3. Wnioski 87
5.3.4. Przykładowe wyniki algorytmów 89
5.3.5. Wyznaczenie parametrów sterujących 93
6. Podsumowanie i kierunki dalszych prac 104
Bibliografia 106
Spis ilustracji 110
Spis tabel 112

Spis skrótów

C2W contour to wrist. Algorytm opierający się na analizie ekstremów lokalnych funkcji odległości kolejnych punktów konturu do punktu nadgarstka;

DT distance transform. Algorytm opierający się o analizę obrazu transformaty odległościowej;

TM template match. Algorytm dopasowania wzorca koła do maski obrazu;

SGONG self-growing and self-organizing neural gas. Algorytm dopasowujący sieć neuronów do obrazu maski dłoni oraz analizujący sąsiedztwo neuronów;

W wrist point. Punkt nadgarstka;

T finger tip. Punkt charakterystyczny klasyfikowany jako czubek palca;

B finger base. Punkt charakterystyczny klasyfikowany jako nasada palca;

hgr hand gesture recognition. Nazwa bazy na której przeprowadzane są testy walidacyjne;

GT ground truth. Punkty wzorcowe;

TP true positive. Punkty prawidłowo wskazane;

TN true negative. Punkty prawidłowo niewskazane;

FP false positive. Punkty niepoprawne wskazane;

FN false negative. Niewskazanie poprawnego punktu;

PPV precyzja testu;

TPR selektywność testu;

F1 miara testu;

I współczynnik jakości wykorzystywany do doboru parametrów sterujących algorytmem autorskim.

Spis symboli i oznaczeń

× symbol wymiaru macierzy lub rozdzielczości obrazu;

***** splot funkcji;

· mnożenie arytmetyczne;

Z(x,y) obraz przedstawiony jako macierz pikseli o wymiarach x × y;

$P(x_P,y_P)$ punkt P o współrzędnych x_P i y_P ;

x_P współrzędna x punktu P;

PQ odcinek o początku i końcu w punkcie P i Q;

|PQ| odległość euklidesowa pomiędzy punktami P i Q;

DT_P element macierzy DT o współrzędnych punktu P;

1. Wstęp

Głównym tematem niniejszej rozprawy doktorskiej jest rozwój technik komunikacji użytkownika z komputerem lub innymi urządzeniami za pomocą komunikatów niewerbalnych, czyli gestów. Przedmiotem badań jest poszerzenie zakresu wiedzy na temat takich sposobów komunikacji, które dla użytkownika będą najbardziej intuicyjne. Naturalny sposób porozumiewania się między ludźmi nazywany jest komunikacją interpersonalną, której definicja encyklopedyczna przedstawiona jest poniżej.

„komunikacja interpersonalna - psychologiczno-socjologiczny proces, dzięki któremu jednostka przekazuje i otrzymuje informacje w bezpośrednim kontakcie z inną osobą; dokonuje się za pomocą przekazu werbalnego (wypowiedzi) i niewerbalnego, na który składają się m.in. mimika, gesty, pozycja ciała, przestrzenne stosunki między komunikującymi się osobami, wysokość i natężenie głosu itp.”[1]

Dzisiejszy rozwój technologii mający na celu poprawę jakości komunikacji skoncentrowany jest głównie na przekazie słownym. Obecnie nie ma najmniejszego problemu, aby błyskawicznie oraz tanim kosztem przesłać przekaz słowny na dalekie odległości za pomocą rozmów telefonicznych, wiadomości tekstowych, poczty elektronicznej lub innych podobnych metod. Zapomina się jednak, iż ważnym składnikiem komunikacji interpersonalnej jest również przekaz niewerbalny, za pomocą którego potrafimy czasami przesłać dodatkowe i ukryte intencje, jakimi mogą być na przykład kłamstwo, sarkazm lub złość. Co więcej, w pewnych sytuacjach przekaz werbalny jest utrudniony. Dzieje się tak w przypadkach przymusu zachowania ciszy, dalekiego dystansu lub zbyt intensywnego hałasu. Istnieją również grupy społeczne, jak na przykład osoby głuchonieme, dla których gesty są jedyną formą komunikacji.

Ponadto, gesty stanowią naturalny sposób komunikacji pomiędzy ludźmi w ich życiu codziennym. Często ludzie używają gestów mimowolnie w celu poprawy zrozumienia ich przekazu lub uzupełnienia przekazywanych treści. Obecnie coraz częściej dostrzega się potrzebę prowadzenia badań nad systemami rozpoznawania gestów, które można stosować między innymi do wideokonferencji [1], telemedycyny oraz wizualizacji w medycynie [2], telerobotyki [3], interaktywnej rzeczywistości rozszerzonej [4], interakcji człowiek-maszyna lub w szczególności człowiek-komputer (ang. human-robot and human-computer interaction, HRI, HCI) [5, 6, 7] i wiele innych [8, 9].

[1] Internetowa encyklopedia PWN. Adres: http://encyklopedia.pwn.pl/haslo.php?id=3924681

1.1. Uzasadnienie konieczności badań

Określenie punktów charakterystycznych dłoni jest jednym z ważniejszych elementów interfejsów wizyjnych. Dokładne określenie lokalizacji punktów charakterystycznych pozwoli na odwzorowanie ułożenia palców a co za tym idzie, na rozpoznanie gestu pokazywanego przez użytkownika. Aktualnie dostępne na rynku komercyjne rozwiązania ograniczają się zaledwie do rozpoznawania kilku wybranych gestów oraz prostego śledzenia ruchu. Co prawda ich funkcjonalność jest uboga, ale wyraźnie uwidaczniają one zapotrzebowanie na rozwój podobnych systemów. Dokładna analiza punktów charakterystycznych pozwoli na rozpoznanie zdecydowanie większych zbiorów gestów, a co za tym idzie znaczne zwiększenie możliwości interfejsów wizyjnych. Takie rozwiązania będą mogły znaleźć wiele zastosowań w dziedzinach związanych z automatyką i robotyką: od przemysłu kontrolerów gier, interaktywnych reklam, poprzez sterowanie manipulatorami na przykład w środowisku sterylnym, nie pozwalającym na dotykanie joysticka, aż do rozwiązań rehabilitacyjnych lub zaawansowanych systemów tłumaczenia języka migowego.

1.2. Układ pracy

Rozdział 1 przedstawia wstęp oraz charakterystykę i klasyfikację gestów. Po opisaniu różnych dziedzin i podziałów gestów, sformułowano założenia dotyczące badań opisywanych w tej pracy. Ostatecznie w podrozdziale 1.6 sformułowana została teza pracy.

Po przeglądzie literatury zaprezentowanym w rozdziale 2, tematyka została ostatecznie zawężona do statycznych gestów dłoni. Rozpatrywane zostaną algorytmy przetwarzania obrazów barwnych, w celu identyfikacji ułożenia dłoni za pomocą lokalizacji punktów charakterystycznych.

Rozdział 2.3 przedstawia przegląd istniejących metod poszukiwania punktów charakterystycznych dłoni. Opisane zostały cztery, wybrane z literatury algorytmy, które zaimplementowano zgodnie z opisem autorów publikacji.

W rozdziale 4 zaproponowany został algorytm, lokalizujący cechy charakterystyczne w obrazie kierunkowym, a następnie klasyfikujący te punkty za pomocą metod sztucznej inteligencji, wnioskowania i drzewa decyzyjnego. Dodatkowo, przedyskutowano metody optymalizacyjne mające na celu skrócenie czasu przetwarzania.

Na etapie walidacyjnym opisanym w rozdziale 5, algorytm został porównany z wybranymi algorytmami z literatury. Testy zostały przeprowadzone we wspólnym środowisku programistycznym, na stworzonej specjalnie do tych celów bazie obrazów.

Podsumowanie oraz propozycja dalszego rozwoju badań znajdują się w rozdziale 6.

1.3. Zakres pracy

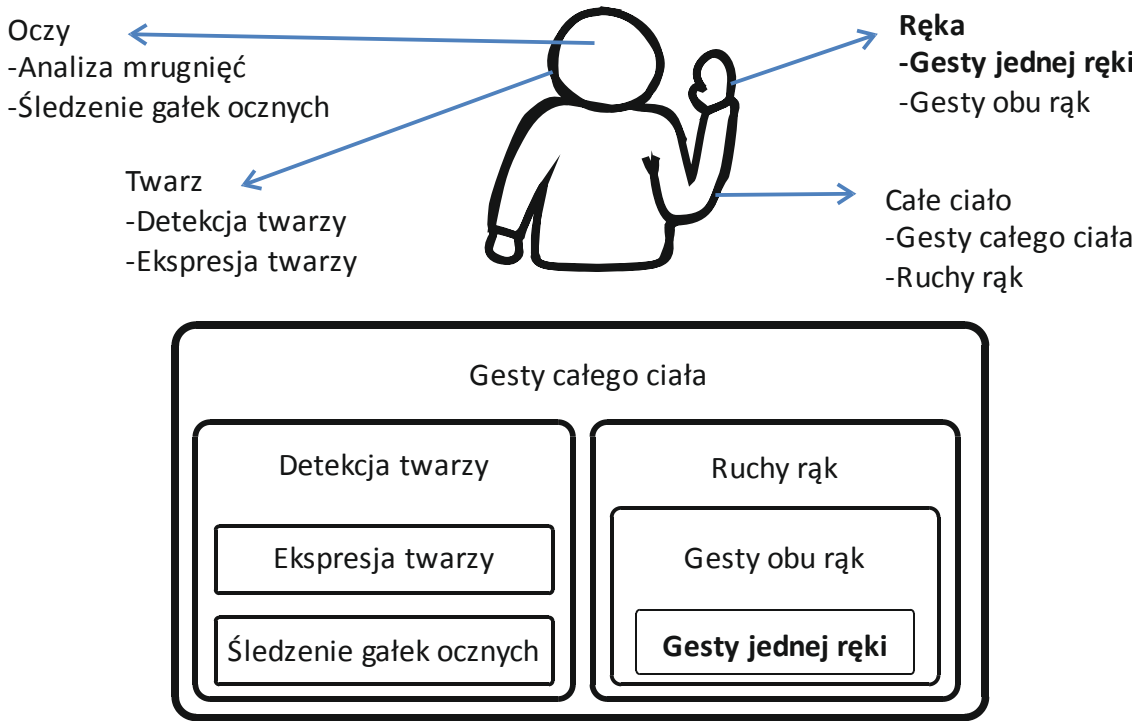
Zagadnienie rozpoznawania gestów jest bardzo obszerne i nie można jednoznacznie mówić o gestach bez odpowiedniego sklasyfikowania problemu oraz bez ulokowania go w jednej z dziedzin. Analizując literaturę wykrywania gestów stworzono trzy kategorie klasyfikujące gesty. Są nimi: obszar zainteresowania, sposoby komunikacji oraz sposób wykonywania gestów.

♦ **Obszar zainteresowania**

Klasyfikacja gestów ze względu na obszar zainteresowania grupuje je w zależności od zbioru części ciała wykorzystywanego w trakcie wykonywania gestu. Rozróżniamy gesty z regionem zainteresowania obejmującym między innymi:

- Ruchy całego ciała;
- Ruchy jednej bądź obu rąk;
- Mimikę twarzy;
- Ułożenie dłoni;
- Mikrogesty.

Dokładna klasyfikacja z uwzględnieniem poszczególnych gestów wchodzących w skład danej grupy przedstawiona jest na Rys. 1.



Rys. 1 Podział gestów ze względu na obszar zainteresowania

Warto zauważyć pewną zależność. Gesty wykonywane za pomocą ruchów rękoma mogą albo zależeć od gestów ułożenia dłoni, jak w przypadku języka migowego, lub mogą tę podklasę ignorować, jak w przypadku komunikacji za pomocą Języka Flagowego [10, 11], w

którym ważna jest tylko pozycja ręki, a samo ułożenie dłoni nie jest znaczące. Opisana powyżej klasyfikacja ma na celu ulokowanie gestów ułożenia dłoni względem pozostałych grup oraz przedstawienie grup z których się wywodzi. Dodatkowo, istnieje możliwość dobrania większej liczby przykładów specjalistycznych algorytmów detekcji wąskiej dziedziny gestów, takich jak mikrogesty [12, 13] zaliczane do ekspresji twarzy lub gestów rąk [14, 15], śledzenie palców [16, 17] zaliczane do gestów jednej ręki, lub analiza wzorców chodu [18] będąca kolejną gałęzią gestów całego ciała.

♦ **Sposób komunikacji**

Gesty można klasyfikować również według sposobu komunikacji. Aby poprawnie zrozumieć dany gest, należy zdefiniować, w jakim języku osoby starają się porozumieć. Wyróżniamy następujące sposoby komunikacji z użyciem gestów:

- Naturalne gesty ciała;

  Zalicza się do nich cały wachlarz gestów, od podświadomych mikrogestów oraz mimiki twarzy, poprzez ruchy dłoni i ciała w trakcie komunikacji, w celu jej usprawnienia, aż do wyuczonych zachowań i postawy w trakcie oficjalnych przemów [19].

- Język migowy;

  Język, który powstał na potrzeby komunikacji pomiędzy oraz z osobami głuchoniemymi. Język ten posiada osobny słownik, w którym każdy gest przypisany jest danemu słowu lub nawet wyrażeniu, czyli fonom lub fonemom. Światowe języki migowe mają odrębną strukturę gramatyczną (nie da się bezpośrednio przetłumaczyć Polskiego Języka Migowego na język polski) oraz nie są jednolite (wyróżniamy na przykład: Polski Język Migowy [20], Angielski Język Migowy [21], Japoński Język Migowy [22]).

- Język palcowy;

  Język, w którym każda litera przedstawiona jest jako odrębne ułożenie dłoni. Istnieją różne alfabety w zależności od kraju pochodzenia. Dla przykładu, litera G w Polskim Alfabecie Palcowym [23] reprezentowana jest inaczej niż w Amerykańskim (American Sign Language) [21].

- Zbiór gestów stworzony na potrzeby sterowania;

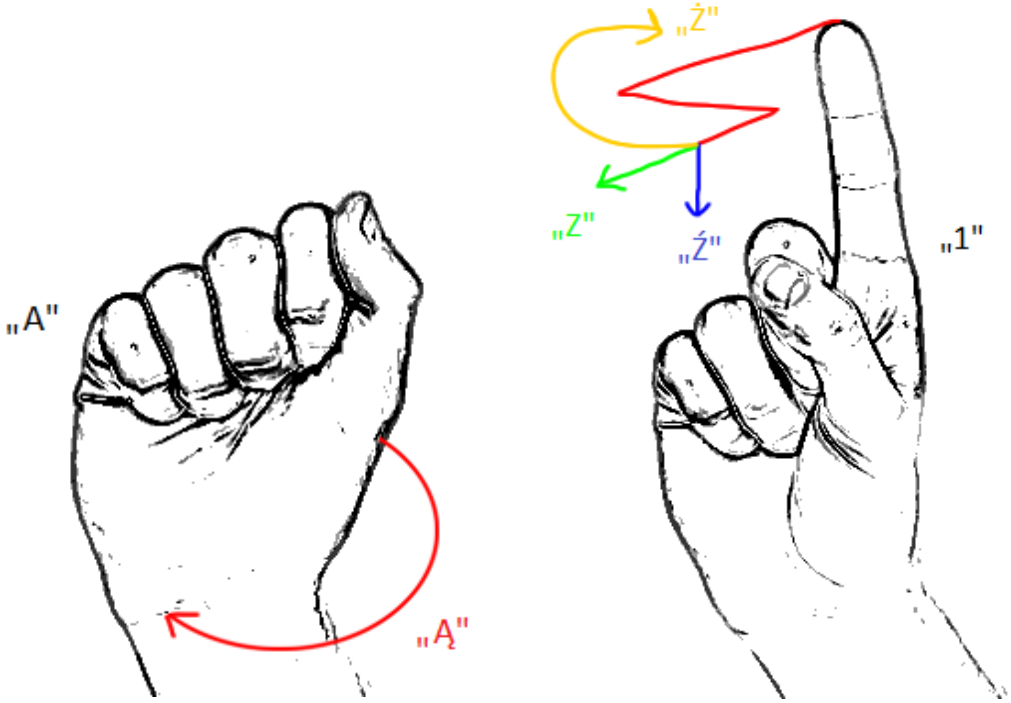
  W ostatnich czasach zauważany jest rozwój urządzeń sterowanych za pomocą gestów. Takie urządzenia sterowane są za pomocą zbioru zadeklarowanych przez programistów gestów. Jest to na przykład gest powiększania obrazu na ekranie dotykowym za pomocą dwóch palców lub gest podniesienia ręki w celu zmiany programu telewizyjnego w inteligentnych telewizorach [24].

- Pozostałe zbiory gestów.

  Istnieje wiele innych sytuacji, w których przekaz werbalny nie jest możliwy, i na potrzeby których stworzono odrębne, mniej lub bardziej usystematyzowane, zbiory gestów. Przykładami są: kierowanie ruchem drogowym, asysta podczas manewru wielkimi maszynami lub komunikacja między nurkami.

♦ **Sposób wykonania gestu**

Ostatni podział gestów opiera się na kryterium sposobu wykonywania gestu. Na ogół, wykonywane gesty mogą mieć charakter pokazania pewnego symbolu lub wykonania pewnej sekwencji ruchowej. W przypadku takiego podziału, gesty dzielimy na:



Rys. 2 Przykłady gestów statycznych oraz dynamicznych.

- Statyczne;

  Podczas wykonywania gestów statycznych liczy się pewne ułożenie ciała. Przykładami mogą być: Litera „A” Polskiego Alfabetu Palcowego, Słowo „Słyszeć” Polskiego Języka Migowego, pokazywana przez dotknięcie palcem wskazującym ucha lub Gest „Stop” pokazywany kierowcy przez policjanta poprzez wzniesienie ręki.

- Dynamiczne.

  Na gest dynamiczny składa się stan początkowy oraz pewna sekwencja ruchu. Przykładami mogą być: litera „Z” Polskiego Alfabetu Palcowego, słowo „Smutny” Polskiego Języka Migowego, pokazywane przez zatoczenie koła pięścią na brzuchu lub gest „Można jechać” prezentowany kierowcy przez policjanta zginającego rękę w

łokciu i utrzymującego dłoń pionowo. Przykładowo, wzniesiony palec wskazujący może zostać zrozumiany jako gest statyczny „1”. Ruch palcem wskazującym, rysujący w powietrzu literę „Z” jest gestem dynamicznym. Aby pokazać gest „Ą” należy wykonać gest „A”, a następnie obrócić nadgarstkiem.

Z punktu widzenia programu komputerowego do rozpoznawania gestów, wyżej wymieniony podział jest bardzo ważny, ponieważ w przypadku gestów statycznych wystarczy analizować obraz przedstawiający dany gest, natomiast w przypadku gestów dynamicznych należy analizować sekwencję obrazów lub film.

1.4. Cel badań

Badania przeprowadzone w niniejszej pracy można sklasyfikować jako analizę gestów statycznych wykonywanych za pomocą dłoni, przedstawiających wybrane znaki z polskiego i amerykańskiego języka palcowego. Na etapie przeglądu literatury zostały wybrane wiodące algorytmy ze wskazanej dziedziny oraz opracowany został autorski algorytm wykrywania punktów charakterystycznych dłoni.

Rozwój każdego z wymienionych obszarów zainteresowania jest równie istotny, ważne jest jednak aby sformułować jasną granicę. Zakres opisywanych badań ograniczono do gestów statycznych jednej dłoni. Wynika to bezpośrednio z wyboru języka komunikacji, czyli alfabetu palcowego. Analiza gestów dłoni została wybrana z powodów szerokiego wachlarzu zastosowań, zwłaszcza z dziedziny sterowania i komunikacji z komputerem oraz innymi maszynami. Dodatkowo, w celu analizy gestów wyższego poziomu, należy poprawnie klasyfikować gesty z najniższego poziomu, czyli wykonywane za pomocą ułożenia dłoni.

Gesty, z alfabetów palcowych, posiadają przejrzystą i konkretnie określoną definicję ułożenia dłoni, co ułatwia stworzenie zbioru gestów potrzebnego do badań. Ujednolicone zasady języka migowego przedstawiają jasną definicję, w jaki sposób należy ułożyć dłoń, aby wykonać dany znak. Osoba przeszkolona z zakresu języka palcowego jest w stanie w jednoznaczny sposób określić jaki gest przedstawiony jest na danym obrazie.

Oparto się na analizie gestów statycznych, ponieważ jest to pierwszy etap analizy ułożenia dłoni. Aby rozpoznawać gesty dynamiczne, należy najpierw opracować niezawodny algorytm rozpoznawania ułożenia dłoni, a następnie śledzić trajektorie wybranych punktów.

Celem badań jest stworzenie algorytmu, który analizuje zdjęcia zawierające dłoń. Dłoń może być widoczna w dowolnym miejscu obrazu oraz z dowolnym ułożeniem i orientacją. Wynikiem algorytmu jest zbiór punktów charakterystycznych za pomocą których można opisać ułożenie tej dłoni.

Głównym założeniem tworzonego algorytmu jest jak największa uniwersalność. Oznacza to, że algorytm powinien działać prawie w każdych warunkach. Parametry minimalnego oświetlenia określane są na poziomie algorytmów detekcji skóry. Rozdzielczość obrazu powinna być większa lub równa 300×300, ponieważ do takiej rozdzielczości skalowane są obrazy w celu ujednolicenia wyników przeprowadzanych badań.

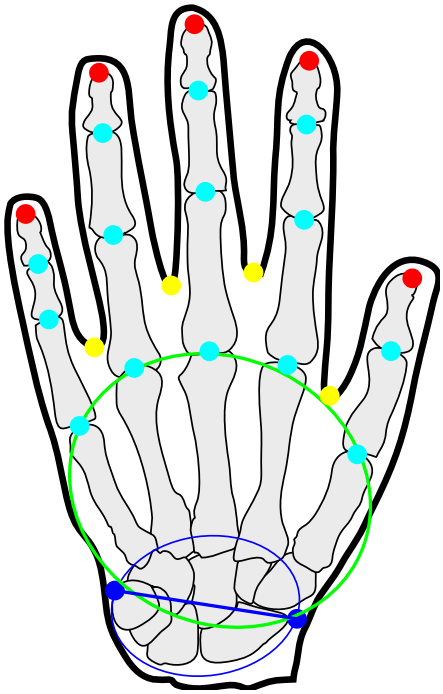
Nie zakłada się personalizacji ani kalibracji algorytmów. Osobiste cechy morfologiczne dłoni różnych użytkowników nie wpływają na działanie opisywanych rozwiązań. Wyjątek stanowią niektóre zmienne algorytmu autorskiego, które wyznaczane są proporcjonalne, za pomocą seryjnych testów opisanych w rozdziale 5.3.5. Na etapie wyznaczania tych parametrów zakłada się, że dłonie różnych użytkowników mogą różnić się wymiarem, jednak proporcje między poszczególnymi członami dłoni pozostają bez zmian.

1.5. Ułożenie i stopnie swobody dłoni

Statyczne gesty wykonywane dłonią polegają na ustalonym ułożeniu części ruchomych dłoni, czyli zgięciu lub wyprostowaniu poszczególnych palców oraz wzajemnym ustawieniu ich w określonej pozycji. Możliwości ruchowe dłoni wynikają bezpośrednio z jej anatomii [25]. Dłoń ludzka posiada 27 kości, wśród których można wyróżnić:

- 5 paliczków dalszych zwanych też dystalnymi;
- 4 paliczki środkowe;
- 5 paliczków bliższych;
- 5 kości śródręcza;
- 8 kości nadgarstka.

Kości każdego palca zaczynają się od nadgarstka i kolejno składają się z kości śródręcza i paliczków połączonych więzadłami. Kciuk nie posiada paliczka środkowego, dlatego składa się z 3 kości, natomiast pozostałe palce składają się z 4 kości każdy.



Rys. 3 Oznaczenie punktów charakterystycznych dłoni: czerwony – czubek palca, opuszka; błękitny – więzadła; niebieski – punkty oraz region nadgarstka; zielony – region śródręcza; żółty – punkty wklęsłe między palcami.

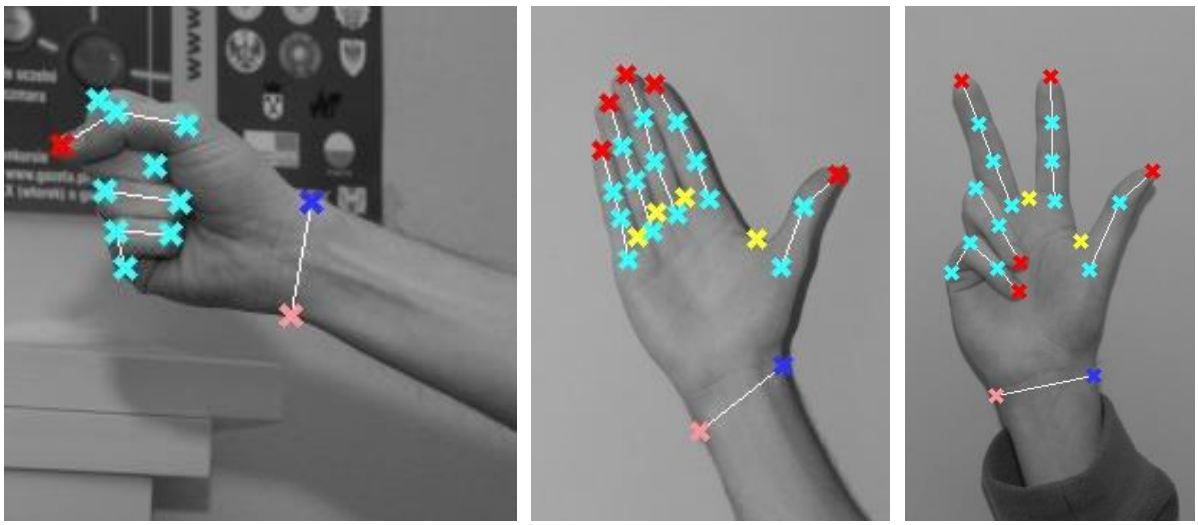
Ciało człowieka zbudowane jest w taki sposób, że kości są elementami stałymi, natomiast więzadła są to ruchome połączenia między kośćmi. Dlatego, aby poprawnie odwzorować ułożenie dłoni, wystarczy zlokalizować początek oraz koniec poszczególnych kości.

Kości nadgarstka nie można rozróżnić na zdjęciu nie będącym zdjęciem rentgenowskim, więc w założeniu nadgarstek traktowany jest jako linia, na której kończą się kości śródręcza.

Kości śródręcza również nie są jednoznacznie widoczne na tradycyjnych zdjęciach, dlatego w trakcie wykrywania wszystkie kości śródręcza traktowane będą jako region śródręcza. Region ten kończy się u podstawy palców.

Każdy palec składa się z dwóch więzadeł oraz trzech kości paliczków. Wyjątkiem jest kciuk, który posiada tylko dwa paliczki i jedno więzadło. Palec zaczyna się więzadłem, które łączy go z kością śródręcza. Ten punkt często nazywany jest podstawą lub nasadą palca. Palec kończy opuszka, nazywana czubkiem palca.

Rozpatrując stopnie swobody, ich liczba zależna jest od liczba członów, czyli kości oraz par kinematycznych, czyli więzadeł. Więzadła mogą być typu czwartego lub piątego, w zależności czy obrót możliwy jest w jednej płaszczyźnie czy w dwóch. Uproszczony model pomijający palec wskazujący oraz traktujący nadgarstek jako unieruchomioną podstawę wyróżnia 16 stopni swobody [26]. Rozszerzając ten model, można wskazać następujące parametry. Każdy palec składa się z 2 par kinematycznych piątego stopnia i jednej pary szóstego stopnia, czyli 4 stopnie na każdy palec. Kciuk posiada o jedną parę mniej, czyli 3 stopnie. Wszystkie kości nadgarstka można przedstawić jako 3 stopnie swobody, traktując łokieć jako podstawę. W sumie daje to 22 stopnie swobody.



Rys. 4 Przykłady ulokowania punktów charakterystycznych

Celem prezentowanych algorytmów będzie jak najdokładniejsze odnalezienie punktów charakterystycznych na obrazach dłoni względem punktów zaznaczonych przez ekspertów. Poprawne odwzorowanie punktów charakterystycznych w przestrzeni dwuwymiarowej pozwala na określenie ułożenia dłoni.

1.6. Teza pracy

W niniejszej rozprawie doktorskiej zaprezentowany jest algorytm mający na celu wykrycie i lokalizacje punktów charakterystycznych dłoni w obrazach cyfrowych. Zadaniem prezentowanego algorytmu oraz algorytmów referencyjnych jest jak najskuteczniejsze wykrycie oraz jak najdokładniejsza lokalizacja wybranych punktów charakterystycznych. Punkty wykrywane są na zbiorze barwnych obrazów cyfrowych przedstawiających dłonie. Przeprowadzenie procedury walidacyjnej posłuży udowodnieniu następującej tezy:

Zaproponowane w pracy metody sztucznej inteligencji polepszają dokładność lokalizacji i detekcji wybranych punktów charakterystycznych dłoni.

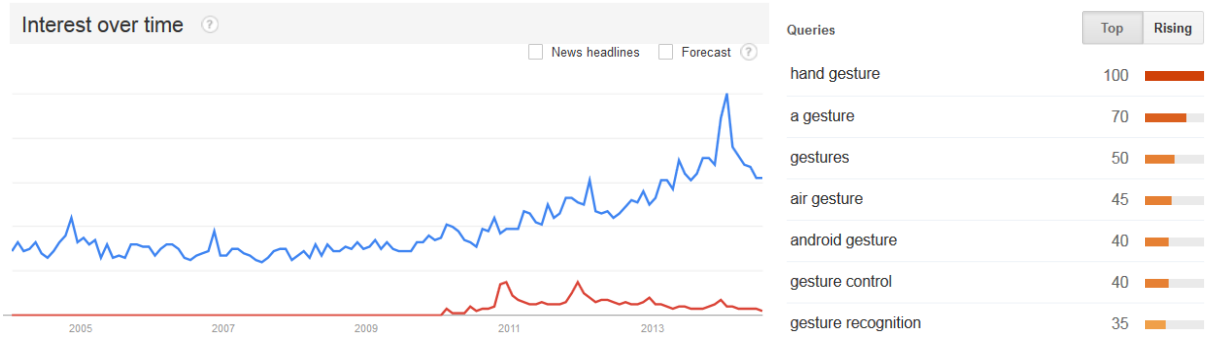
Do celów porównania autor wybrał i zaimplementował cztery algorytmy opisane w literaturze. Algorytmy te przedstawione są w rozdziale 2.3.

W podejściu prezentowanym w pracy Autor zaproponował algorytm hybrydowy łączący klasyczne metody przetwarzania obrazów z procesami decyzyjnymi, opartymi na drzewach decyzyjnych, zawierającymi oryginalne reguły wnioskowania.

Algorytmy przetwarzania obrazu wykorzystywane są w celu utworzenia zbioru punktów charakterystycznych na obrazach. Analiza obrazu z użyciem algorytmów sztucznej inteligencji służy do wyselekcjonowania i sklasyfikowania wybranych punktów charakterystycznych dłoni. Zaproponowany algorytm poszukiwania punktów charakterystycznych cechuje się lepszą dokładnością zarówno pod względem dokładnej lokalizacji jak i odpowiedniej detekcji.

2. Analiza stanu wiedzy

Tematyka rozpoznawania gestów w ostatnich latach cechuje się dużą popularnością. Zaobserwować to można na Rys. 5, przedstawiającym wykres popularności poszukiwania frazy „gest” (ang. gesture), jako dziedziny nauki. Najczęściej słowo „gest” pojawiało się wraz z wyrażeniem „gest dłoni” (ang. hand gesture). Co więcej, zainteresowanie tematyką gestów rośnie od 2011 roku, kiedy to na rynek wszedł Microsoft Kinect [27], relatywnie tania kamera odległościowa z zaimplementowanymi algorytmami detekcji sylwetki człowieka, używana do sterowania grami wideo. Zainteresowanie społeczeństwa możliwościami Kinecta przyczyniło się do rozwoju podobnych technologii, takich jak Samsung Smart Interaction [28], wykorzystywanej od około 2012 roku w telewizorach Samsung Smart TV w celu sterowania telewizorem za pomocą gestów. W tym samym czasie firma Samsung wprowadziła również możliwość sterowania telefonem komórkowym z serii Galaxy gestami nazywanymi Air Gesture [29].



Rys. 5 Wykresy popularności wyszukiwania słów "gesture" (niebieska linia) oraz "kinekt"(czerwona linia) z użyciem wyszukiwarki Google, stworzone przez Google Trends [30]. Po prawej zaprezentowano frazy, w których najczęściej pojawiało się słowo „gesture”.

Osiągnięcie skuteczności pozwalającej na pierwsze wdrożenia spowodowało ogólną dostępność urządzeń typu Kinect. Ponadto fascynacja gestami kreowana przez firmy wprowadzające te produkty wpłynęła na wzrost zainteresowania tematyką wykrywania gestów, zwłaszcza gestów ciała, rąk i dłoni z dziedziny gestów wykrywanych na potrzeby sterowania.

Pierwszym krokiem mającym na celu stworzenie algorytmu rozpoznawania gestów jest poprawne określenie dziedziny badań, dostępnych rozwiązań oraz możliwych metod implementacyjnych.

W niniejszym rozdziale dokonano przeglądu literatury oraz wyszczególniono rodzaje i systemy rozpoznawania gestów. Przedstawiono znane, przebadane i zaimplementowane rozwiązania z wyjaśnieniem założeń, które zostały przyjęte. Wytłumaczono także wytyczne, które powinien spełniać system, mający na celu wykrywanie i lokalizacje punktów charakterystycznych dłoni w obrazach cyfrowych.

Po wytłumaczeniu powodów zastosowania systemów wizyjnych, w podrozdziale 3.2 opisano wybrane metody analizy obrazu, które będą wykorzystywane podczas walidacji w opisywanych badaniach.

2.1. Przegląd literatury

Głównym założeniem niniejszej rozprawy jest zatem badanie algorytmów detekcji ułożenia dłoni. Zaproponowano podział przeglądu literatury dzieląc systemy detekcji gestów ze względu na sposób akwizycji danych. Są to systemy sensoryczne, które opierają akwizycje danych na różnego rodzaju czujnikach bez użycia kamer, systemy wizyjne światła widzialnego oparte o tradycyjne kamery i pozostałe systemy wizyjne, w których wykorzystywane są kamery inne niż światła widzialnego.

2.1.1. Systemy sensoryczne, niewizyjne oraz systemy wizyjne wspomagane

Mimo dużego zainteresowania tematyką gestów z okresu ostatnich czterech lat, wykrywanie gestów dłoni było wyzwaniem naukowców od znacznie dłuższego czasu. Nawet przed popularyzacją aparatów cyfrowych, które ukierunkowały badania na użycie obrazów cyfrowych, pierwsze podejścia do akwizycji danych pozwalających na odwzorowanie ułożenia dłoni tworzone były w oparciu o specjalistyczne nakładane czujniki. W ten sposób powstała rękawica, która zakładana na rękę odwzorowywała kąty zginania palców. P. Halarnkar [31] opisuje rozwiązania z użyciem rękawicy sensorycznej nazywając je CyberGlove oraz DataGlove, powstałych już w okolicach 1989 roku. CyberGlove jest to rękawica, która składa się z popękanych światłowodów, przewodzących mniej światła w zależności od stopnia zgięcia. DataGlove natomiast składa się z potencjometrów, zmieniających swoją rezystancję wraz ze zmianą kąta nachylenia. Niestety to nazewnictwo nie jest jednoznaczne, i tak O. Luzanin [32] wszystkie rozwiązania określa jako CyberGlove, natomiast w publikacjach [33, 34, 35, 36, 37] wspomniane rozwiązania nazywane są DataGlove, dla opisania różnych rozwiązań przewodowych lub bezprzewodowych wykorzystujących czujniki tensometryczne, rezystancyjne lub akcelerometry.

Rękawice są najbardziej popularnym rozwiązaniem. Powstają jednak urządzenia odmienne. Jedno z rozwiązań przedstawia pierścień, który posiada wbudowany akcelerometr i inne czujniki, zdolne do śledzenia ruchu palca w powietrzu oraz sterowania urządzeniami użytku domowego [38]. Istnieją rozwiązania śledzenia palców nad powierzchniami dotykowymi za pomocą wiązek laserów [39] oraz połączenia akcelerometrów z czujnikami laserowymi [40].

Dużą zaletą systemów niewizyjnych opartych na czujnikach jest możliwość doboru odpowiednich podzespołów do konkretnego zadania. Dzięki temu uzyskuje się dużą precyzję oraz skuteczność w wykonywaniu konkretnego zadania, dla którego urządzenie zostało stworzone. Niestety rozwiązania takie nie są uniwersalne. Oznacza to, że aby system działał, potrzebne jest to właśnie specjalistyczne urządzenie, które często jest bardzo kosztowne. Dlatego w ramach prowadzonych badań, ostatecznie postanowiono odejść od rozwiązań sensorycznych, z założeniem stworzenia systemu opartego o znacznie tańsze i powszechne kamery światła widzialnego, szeroko dostępne w postaci kamer internetowych lub aparatów w telefonach komórkowych.

Istnieją jednak systemy wizyjne, które działają z użyciem kamer, natomiast zaimplementowane algorytmy wspierane są przez detekcję oznaczonych kolorów lub specjalnych znaczników przyczepianych na ciało. Przykładami mogą być rozwiązania oparte na algorytmach poszukiwania kolorów na barwnej rękawicy, w których palce pokolorowane są w charakterystyczny sposób [41], wykrywanie kolorowych tasiemek obwiązanych wokół palców [42] lub popularny kontroler gier Playstation Move [43], który wysyła sygnał do kamery podczerwieni umieszczonej w punkcie referencyjnym. Niestety, mimo że rozwiązania te są tańsze od zaawansowanych systemów sensorycznych, to nadal wymagają od użytkownika, aby posiadał specjalne urządzenia lub markery na swoim ciele.

Rozwijając pomysł śledzenia znaczników, stworzono laboratoria Motion Capture. Za pomocą wielu kamer, system jest w stanie lokalizować dokładnie wyróżniające się znaczniki, które umiejscowione są na ciele człowieka. Wykryte punkty lokalizowane są w przestrzeni trójwymiarowej tworząc wzorce chodu i wzorce zachowań używane do modelowania postaci w grach i animacjach [44].

W trakcie badań nawiązano krótką współpracę z laboratorium Motion Capture, należącego do Polsko-Japońskiej Wyższej Szkoły Technik Komputerowych - Wydziału Zamiejscowego Informatyki w Bytomiu, jednak okazało się, że śledzenie punktów charakterystycznych dłoni wymaga zbyt dużej precyzji. Ponadto laboratorium takie mogłoby przydać się do stworzenia bazy wzorcowej gestów dynamicznych, które na tym etapie badań nie są analizowane.

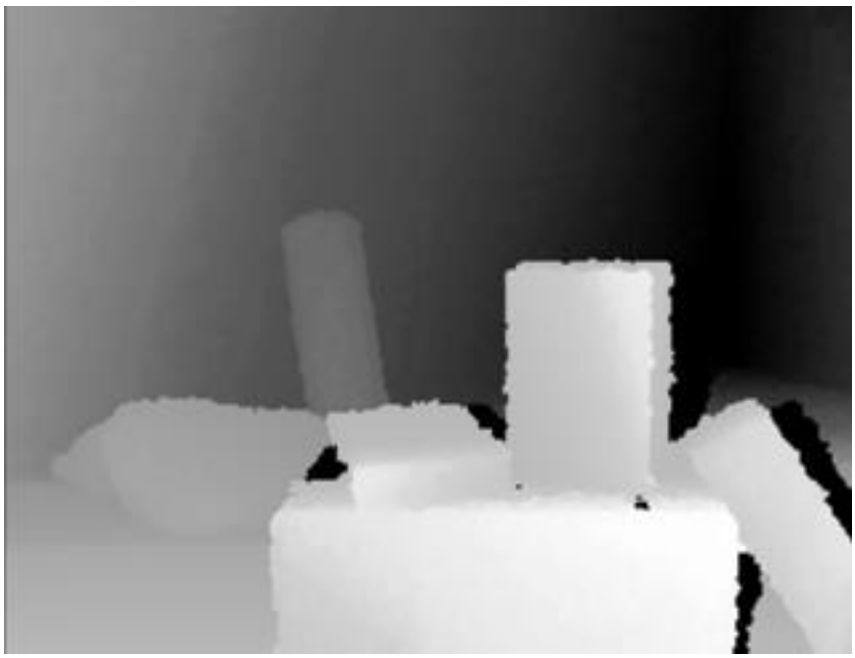
2.1.2. Systemy wizyjne oparte o kamery inne niż światła widzialnego

Tradycyjna kamera światła widzialnego wykrywa składowe wiązki światła odpowiadające trzem podstawowym kolorom. Obraz wyjściowy przenosi bezpośrednią informacje o widzianych kolorach oraz natężeniu światła. Osoba stojąca przed kamerą nie jest w żaden prosty sposób inna od tła, na którym stoi. Dlatego często korzysta się z kamer innych niż kamery światła widzialnego, przekazujących dodatkowe informacje o poszukiwanym obiekcie. Za pomocą przedstawianych danych można w bezpośredni sposób oddzielić obiekt poszukiwany od tła.

Jednym z najpopularniejszych rozwiązań jest kamera odległościowa Kinect, wprowadzona na rynek przez firmę Microsoft, będąca kontynuacją projektu ZCam firmy 3DV Systems. Kamera ta składa się przede wszystkim z promiennika podczerwieni oraz kamery podczerwonej. Kamera z filtrem podczerwonym rejestruje siatkę punktów wyświetlanych przez promiennik. Następnie określana jest odległość rzutowanych punktów od kamery, za pomocą której tworzona jest mapa głębi, przedstawiona na Rys. 6. Obraz ten nie jest idealnym odwzorowaniem głębi, ponieważ występuje efekt paralaksy, przedstawiony jako czarny cień na obrazie. Ponadto kamera nie jest wystarczająco czuła, aby dokładnie opisać obiekty. Na Rys. 6 widać, że kontury są poszarpane, a po dokładniejszym zbadaniu można zauważyć, że odległość przestrzeni płaskich nie jest jednolita. To powoduje, że taka kamera jest dobra do odseparowania sylwetki człowieka od tła, jednak nie jest dostatecznie precyzyjna, aby wykryć dokładne ułożenie dłoni.

Dostępność Kinecta zaowocowała szeregiem badań z użyciem tej kamery. Są to głównie badania związane z detekcją sylwetki [45] oraz gestów rąk [46], jednak zdarzają się też publikacje opisujące detekcje prostych gestów ułożenia dłoni [47].

Szeroko dostępna i relatywnie tania kamera Kinect sprawdza się w prostych grach, jednak nie jest tak dokładna jak profesjonalne kamery nazywane „time of flight camera". Są to urządzenia pozwalające na tworzenie map głębi na podstawie pomiaru prędkości światła, czyli czasu przelotu (ang. time of flight, ToF) strumienia światła dla każdego punktu w przestrzeni czujnika. Kamery takie są znacznie droższe, ale są też w stanie zapewnić odpowiednią jakość potrzebną do wykrycia gestów dłoni [48, 49]. Powstają również rozwiązania łączące mapę odległościową z kamerą światła widzialnego [50].



Rys. 6 Przykład mapy odległościowej wygenerowanej przez Kinect

Wiązka światła podczerwonego wykorzystywana jest również w kamerach termowizyjnych, które także zapewniają łatwą segmentację obrazu, wyszczególniając sylwetkę człowieka z otoczenia za pomocą różnic temperatur. Przykładem jest publikacja opisująca zastosowanie rozwiązania wykorzystującego kamerę termowizyjną oraz płachtę odbijającą sygnaturę cieplną do wyliczenia pozycji sylwetki człowieka oraz detekcji gestów trójwymiarowych [51]. Można również odnaleźć prostsze zastosowania, jak na przykład śledzenie sygnatur cieplnych palców przesuwanych po blacie [52] lub skanowanie powierzchni wiązką podczerwoną w celu wykrycia miejsc styczności palców z biurkiem [53].

Rozwiązania korzystające z czujników podczerwonych lub innych specjalistycznych kamer posiadają dużą zaletę łatwej segmentacji sylwetki człowieka od tła. Jednak w celu poprawnego rozpoznawania ułożenia dłoni wymagana jest duża szczegółowość obrazu. Jednakże cena kamer bardzo szybko rośnie wraz ze wzrostem ich dokładności i rozdzielczości. Ponadto konieczność kalibracji tych kamer może spowodować, że rozwiązanie będzie działało jedynie w warunkach laboratoryjnych.

2.1.3. Systemy wizyjne oparte na kamerach światła widzialnego

Najbardziej uniwersalnym i najpopularniejszym podejściem do wykrywania gestów jest użycie tradycyjnych kamer światła widzialnego. Są one szeroko dostępne, często dołączane do urządzeń takich jak laptopy, kamery i telefony komórkowe. Co więcej, cena specjalistycznych aparatów fotograficznych i kamer jest znacznie niższa od porównywalnej klasy kamer podczerwonych czy odległościowych.

Przy okazji systemów opartych na kamerach innych niż światła widzialnego, omówiono główny problem związany z cyfrowym obrazem kolorowym. Mianowicie obraz taki przedstawia nasycenie wiązki światła w danym punkcie przestrzeni. W bezpośredniej interpretacji można rozróżniać obiekty, które mają różny kolor lub różne nasycenie światła. Natomiast człowiek stojący na tle niekontrolowanego otoczenia nie jest w bezpośredni sposób odróżnialny. Dopiero algorytmy przetwarzania obrazu są w stanie wyłonić cechy charakterystyczne pozwalające na detekcję obiektu zainteresowania.

Standardowe podejście do wyłonienia obiektu wykorzystywane przy przetwarzaniu obrazów cyfrowych dzieli się na 3 etapy [54, 55]:

1. Obróbka wstępna – eliminacja zakłóceń, wydobywanie obiektów z tła oraz ich lokalizacja na przestrzeni obrazu;
2. Obróbka dalsza – wyłanianie cech charakterystycznych wykrytych wcześniej obiektów;
3. Analiza – klasyfikacja obiektu za pomocą wyznaczonych cech i wykorzystywanie wyników klasyfikacji np. do sterowania.

W przypadku algorytmów wykrywania gestów, pierwszym etapem, nazywanym obróbką wstępną, jest segmentacja obszaru dłoni. W celu odnalezienia dłoni najczęściej wykorzystuje się algorytmy wykrywania regionu skóry w obrazie. W przeglądzie algorytmów detekcji skóry [56] wyróżnia się podziały metod w zależności od przestrzeni barw oraz sposobów klasyfikacji pikseli. Z metod klasyfikacji pikseli wyróżnia się progowanie kanałów danej przestrzeni barw stałymi wartościami oraz użycie modelu pikseli skóry, zbudowanego za pomocą zbioru treningowego. W celu zbudowania wiedzy odnośni skóry Wyróżnia się modele parametryczne, nieparametryczne oraz dynamiczne. Przykładami bardziej zaawansowanych algorytmów detekcji regionu występowania skóry są: algorytmy propagacyjny [57] lub wykorzystujący model przestrzenny [58].

W przypadku statycznych obrazów barwnych o niekontrolowanym tle, wykrywanie skóry jest najbardziej popularną metodą segmentacji bez użycia znaczników. Ewentualną konkurencję stanowią metody poszukiwania cech charakterystycznych, jednak należą one do etapu drugiego, omijając poniekąd etap pierwszy.

Obróbka wstępna jest bardzo ważnym etapem systemu wizyjnego, często w dużym stopniu wpływająca na ostateczny wynik. To właśnie w celu poprawnej i łatwej segmentacji regionu dłoni lub człowieka od tła, wykorzystuje się systemy wizyjne oparte o kamery inne niż światła widzialnego. W algorytmach skupiających się na obróbce dalszej, czyli ekstrakcji cech charakterystycznych, etap pierwszy jest pomijany lub nakłada się na niego wiele

ułatwiających segmentację założeń. Stosowane jest na przykład znane tło, które następnie odejmowane jest od obrazu, w celu ustalenia regionu, w którym znajduje się użytkownik [59].

W niniejszej pracy w trakcie badań stosuje się maski dłoni z regionem skóry oznaczonym przez ekspertów w celu pominięcia błędu generowanego na etapie detekcji skóry. Algorytm jednak zaimplementowany jest we wspólnym środowisku wraz z algorytmami wstępnego przetwarzania obrazu [57, 58], tak aby mógł działać na obrazach o niekontrolowanym otoczeniu.

Z dziedziny kamer światła widzialnego można wyszczególnić jeszcze dwa warte uwagi rozwiązania. W dziedzinie mikrogestów należy analizować szybkie, niekontrolowane skurcze mięśni. Do tych celów używa się kamery szybkiej, która potrafi uchwycić tysiące klatek na sekundę. Tradycyjna kamera pobiera 25 - 30 klatek na sekundę. Danych jest natomiast tak wiele, że trudna jest ich pełna akwizycja i przetwarzanie w czasie rzeczywistym [13]. Kolejne rozwiązanie bazujące na połączeniu obraz z dwóch kamer tworzy obraz stereowizyjny, który następnie na podstawie dopasowania cech charakterystycznych można przekształcić w mapę odległościową. Dzięki temu możliwa jest łatwa segmentacja sylwetki człowieka od otoczenia [60].

Jeden z głównych podziałów gestów wyodrębnia gesty statyczne oraz dynamiczne, gdzie gesty statyczne opierają się na analizie obrazu, a gesty dynamiczne na analizie sekwencji wideo. Analiza wideo opiera się o śledzenie wykrytych punktów charakterystycznych oraz interpretację ścieżki tych ruchów. W przypadku śledzenia wykorzystywane są algorytmy, na przykład wykorzystujące przepływ optyczny (ang. optical flow) [61], który również może być zastosowany dla kamer ToF [62], lub algorytm SURF [63, 64, 65]. Jeżeli chodzi o śledzenie ścieżki, najczęściej wykorzystywaną metodą jest metoda Ukrytych Modeli Markowa [8, 36, 37, 59, 60, 62, 63], jednak napotkano również metodę wykorzystującą HCRF (ang. hidden conditional random fields) [64].

2.2. Założenia dotyczące zakresu badań

Po przeanalizowaniu różnych rozwiązań, ostatecznie wybrano następujące założenia odnośnie prezentowanego algorytmu analizy gestów.

Przede wszystkim badania prowadzone są z użyciem stacjonarnych obrazów cyfrowych pozyskanych z kamer światła widzialnego. Takie rozwiązanie gwarantuje możliwie tanie i uniwersalne stworzenie ostatecznego produktu.

Algorytm analizy cech charakterystycznych powinien pracować w czasie rzeczywistym, aby algorytm dało się rozszerzyć o moduł śledzący ruchy punktów charakterystycznych do celów detekcji gestów dynamicznych.

Nie zakłada się żadnych ograniczeń odnośnie obrazów. Oznacza to, że dłoń na obrazie może mieć dowolne tło, orientację oraz dowolną długość rękawa.

Algorytm zawiera metody przetwarzania obrazów oraz wykrywanie i lokalizacje punktów charakterystycznych dłoni, takich jak punkty nadgarstka, czubki i nasady palców.

2.3. Przegląd istniejących metod wykrywania punktów charakterystycznych

Głównym celem algorytmów wykrywania ułożenia dłoni jest jak najdokładniejsze odwzorowanie jej ułożenia. Istniejące rozwiązania można zaklasyfikować do dwóch grup: algorytmy dopasowujące model dłoni do zaobserwowanego stanu oraz algorytmy poszukiwania punktów charakterystycznych z użyciem metod ekstrakcji cech z obrazu [66].

W przypadku podejścia pierwszego, tworzony jest model dłoni z odwzorowaniem kinematyki oraz możliwości ruchowych wszystkich członów. Model dłoni przyjmuje różne stany w zależności od parametrów nim sterujących. Następnie zaobserwowany stan z systemu wizyjnego [67] lub sensorycznego [68] jest porównywany z możliwymi ułożeniami modelu. Na podstawie stanu modelu tworzona jest informacja o stanie ułożenia dłoni.

Drugi zestaw metod ustalania ułożenia dłoni opiera się na odnajdywaniu cech charakterystycznych z użyciem przetwarzania obrazu. Danymi wejściowymi do programu wykrywającego i lokalizującego punkty charakterystyczne są obrazy dłoni. Wyjściem natomiast jest zbiór wykrytych punktów charakterystycznych wraz z ich współrzędnymi, czyli lokalizacją. Dodatkowo, niektóre algorytmy klasyfikują punkty, ustalając przynależność do poszczególnej grupy punktów. W pełni automatyczny program do rozpoznania ułożenia dłoni składa się z następujących etapów:

1) Wykrycie regionu skóry i stworzenie maski dłoni
2) Wykrywanie i lokalizacja punktów charakterystycznych
3) Klasyfikacja wykrytych punktów w celu ustalenia ułożenia dłoni

Celem etapu pierwszego jest oddzielenie regionu zainteresowania, czyli dłoni od tła. W tym celu najczęściej stosuje się algorytmy detekcji regionu skóry. Dokładny opis rozwiązań segmentacji obrazu, wykorzystywanych metod oraz stosowanych czujników przedstawiony został w rozdziale 2.1.

W literaturze można spotkać wiele różnych algorytmów rozpoznawania punktów charakterystycznych, w których główny nacisk położony jest na etap wykrywania i lokalizacji punktów charakterystycznych. Można odnaleźć rozwiązania bazujące na morfologii obrazu [69], analizy punktów miejsc przegięć oraz wklęsłości krzywej opisującej konturu dłoni [70], analizy cech Haara [71] lub rozwiązania wykorzystujące analizę miejsc wykrycia skupisk pikseli (ang. blobs) [71, 72].

W kolejnej części rozdziału przedstawiono wybrane algorytmy wykrywania punktów charakterystycznych. Wybrano cztery algorytmy, których dokładny opis znajduje się w odpowiednich podrozdziałach. Dodatkowo, opis i porównanie tych metod można odnaleźć w publikacji [73].

Ważnym etapem w ustaleniu regionu poszukiwań oraz orientacji dłoni jest wykrycie nadgarstka na obrazie. Większość opisywanych rozwiązań skupia się na wykrywaniu punktów charakterystycznych zakładając, że punkt nadgarstka jest znany lub jest w ustalonym miejscu obrazu [22, 72, 74]. Publikacje, poświęcone jedynie wykrywaniu nadgarstka lub takie, w

których detekcja nadgarstka jest dokładnie przebadana są rzadkością [75, 76, 77]. Oczywiście zdarzają się również metody, które nie wymagają wyznaczenia punktu nadgarstka [71, 78].

Ostatni etap algorytmów detekcji punktów charakterystycznych polega na klasyfikacji wykrytych punktów. Jego celem jest sformułowanie opisu wykrytych elementów, oraz usystematyzowanie tego opisu w taki sposób, aby łatwo można było wykorzystać te dane, w zależności od ostatecznego zastosowania programu. W przypadku, gdy na przykład końcowy produkt miałby być stosowany do rozpoznawania języka migowego, zbiór wykrytych punktów charakterystycznych porównywany byłby do ułożenia dłoni alfabetu palcowego.

2.3.1. Analiza odległości punktów konturu od nadgarstka

N. Tanibata w swoim artykule [22] opisuje wiele etapów prowadzących do powstania systemu rozpoznawania Japońskiego Języka Migowego, pokazywanego przez osobę siedzącą przed kontrolowanym tłem. Temat rozpoznawania samego ułożenia dłoni nie jest ani obszernie opisany ani dostatecznie zbadany. W rozdziale 4 tego artykułu autor używa zaledwie dwóch zdań do opisu metody poszukiwania punktów charakterystycznych, które w dokładnym tłumaczeniu brzmią: liczba palców ustalona jest jako liczba lokalnych maksimów funkcji odległości między nadgarstkiem a punktami konturu regionu ręki. Pozycja nadgarstka ustalona jest jako punkt regionu dłoni znajdujący się najbliżej łokcia [22].

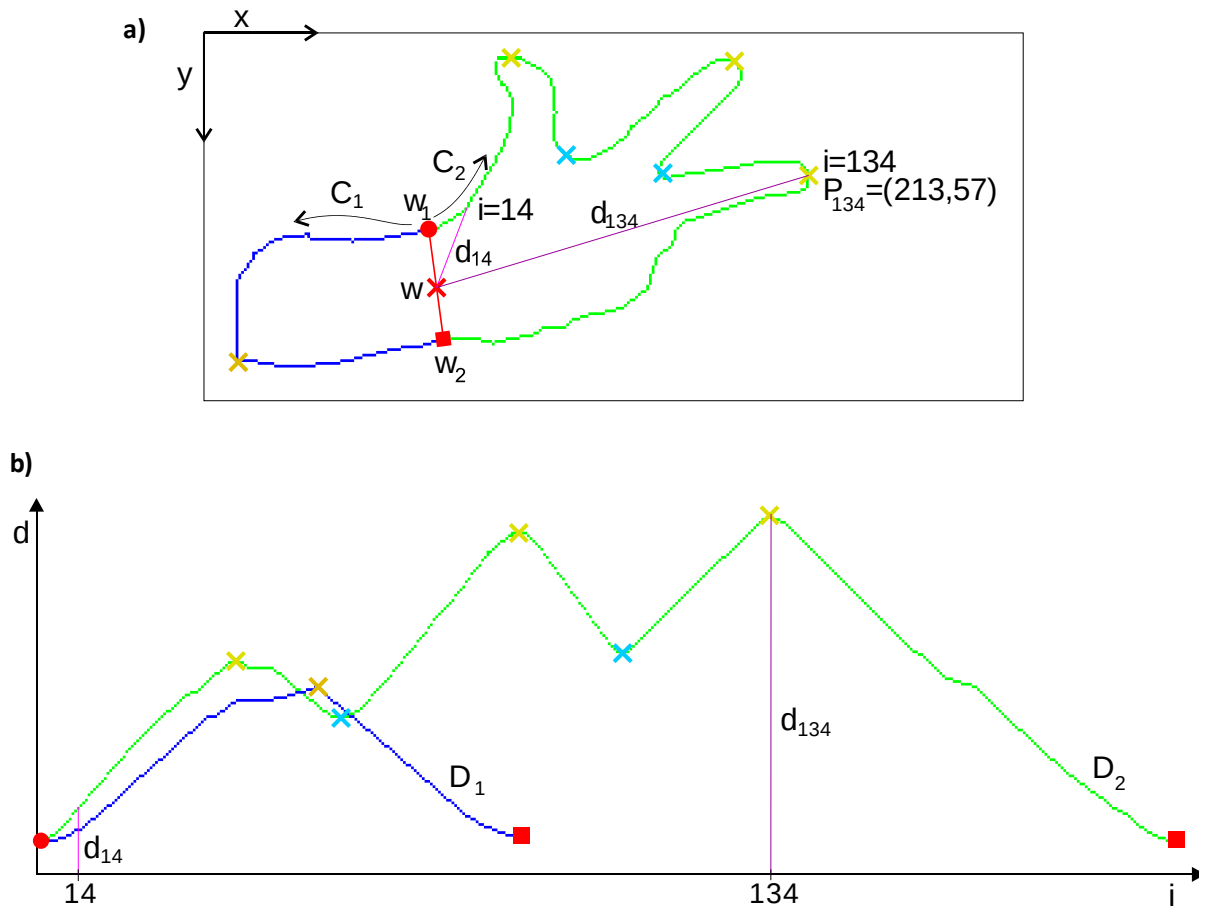
Analizując publikację można wyciągnąć dodatkowe wnioski odnośnie wielu założeń jakie dokonano, czyli między innymi: kontrolowane tło, siedząca postawa osoby, statyczna kamera i długi rękaw pozwalający na dokładniejsze ustalenie regionu skóry. Powyższa metoda, mimo wielkich uproszczeń i bardzo zwięzłemu opisowi sprawdzała się w takim systemie. Po kilku modyfikacjach metoda ta posiada również potencjał do zastosowania w rozwiązaniach poszukiwania punktów charakterystycznych w otoczeniu niekontrolowanym.

Aby pozbyć się ograniczeń wynikających z opisywanych założeń, należy przede wszystkim należy przeprowadzić odpowiednią segmentację, na przykład z zastosowaniem algorytmów wykrywania skóry opisywanych w rozdziale 3.2.1. Również poszukiwanie nadgarstka nie jest sprawą łatwą, jeśli użytkownikowi nie nałoży się ograniczeń odnośnie długości rękawa. Do takiego przypadku opracowano algorytm poszukiwania nadgarstka opisany w rozdziale 4.1, który następnie został dołączony do tej metody. W przypadku kiedy znany jest kontur dłoni oraz określone zostały punkty nadgarstka można przystąpić do wyliczenia funkcji odległości zgodnie z propozycją N. Tanibata oraz do lokalizacji punktów charakterystycznych.

Po niezbędnych modyfikacjach, wyliczenie funkcji odległości odbywa się w sposób następujący. Po etapie odnalezienia punków nadgarstka W_1 oraz W_2, odnajdywany jest punkt środkowy W. Następnie kontur maski zostaje podzielony na 2 części, C_1 oraz C_2, które rozgraniczają punkty nadgarstka. C_1 oraz C_2 są to ciągi punktów zawierające współrzędne kolejnych punktów konturu. Znając ciąg C, można wyznaczyć ciąg D, zawierający odległości d pomiędzy współrzędnymi kolejnych punktów $c(x_c,y_c)$ z ciągu C oraz współrzędnymi punktu nadgarstka $w(x_w,y_w)$. Ciąg D wyznaczany jest korzystając ze wzoru na odległość dwóch punktów:

$$d_i = \sqrt{\left(\mathrm{x}_{c_i} - \mathrm{x}_\mathrm{w}\right)^2 + \left(y_{c_i} - \mathrm{w}_\mathrm{y}\right)^2}\,. \qquad (1)$$

Zmienna *i* oznacza zmienną kroczącą, wskazującą na kolejne wartości w ciągach *D* i *C.* Ciąg *D* wyliczany jest dla obu części konturu rozdzielonego punktami nadgarstka, tak że ostatecznie znane są D_1 i D_2 dla C_1 i C_2. Ciągi te prezentowane są na Rys. 7 jako funkcja dyskretna. Zaznaczono je odpowiadającymi kolorami, niebieskim (C_2) oraz zielonym dla (C_1).



Rys. 7 Przykład działania algorytmu analizy odległości punktów konturu od nadgarstka. a) Kontur obrazu źródłowego, b) funkcja odległości kolejnych punktów konturu od punktu nadgarstka

Przykładowo dla czternastego elementu konturu C_1 odległość została wyliczona oraz naniesiona na wykresie w punkcie *i=14*. Dzięki takiej reprezentacji można zastosować algorytm poszukiwania minimum lokalnego, w taki sposób, aby odnalazł wszystkie minima oraz maksima lokalne. Jako wynik takiego kroku uzyskane zostaną punkty zaznaczone na wykresie żółtymi oraz niebieskimi krzyżykami. Na przykładzie z Rys. 7 widać dwa ciągi zawierające punkty konturu:

$$\begin{aligned} C_1 &= \{w_1, (73,64), (74,63), (75,62), \ldots, c_{134} = (213,57), \ldots, w_2\} \\ C_2 &= \{w_1, (71,65), (70,65), (69,66), \ldots, w_2\}\,. \end{aligned} \qquad (2)$$

Jeżeli maksimum lokalne znajduje się w *134* elemencie konturu C_1, odczytane zostaną współrzędne tego punktu, czyli punkt *(213,57)*. W taki sposób odnajdywane są punkty czubków palców jako lokalne maksima oraz punkty między palcami jako lokalne minima.

W następnym kroku należy rozstrzygnąć, który kontur zawiera region dłoni, a który region nadgarstka, łokcia lub części ręki. Zaobserwowano, że dłoń znajduje się po tej stronie nadgarstka, po której kontur jest bardziej „postrzępiony", czyli będzie posiadał więcej ekstremów lokalnych. Tę prawidłowość można zaobserwować na Rys. 7, gdzie dla funkcji D_2 wykryto tylko jedno maksimum lokalne. W przypadkach, gdy ilość ekstremów lokalnych jest identyczna dla obu funkcji, wybierany jest ten ciąg, dla którego suma różnic między minimami a maksimami lokalnymi jest większa.

Zaimplementowany algorytm wykazuje się dużą skutecznością. Z dużą dokładnością odnajdywane są punkty czubków palców wyraźnie wzniesionych. W przypadku gdy palce są połączone, różnice między ekstremami lokalnymi nie są duże, jednak czasem wystarczające, aby wykryć czubki palców. Algorytm jednak podatny jest na zafałszowania wynikające głównie z postrzępionego konturu lub błędnego wykrycia punktu nadgarstka. Drobną wadą jest również wykrywanie punktów czubków palców na konturze, a nie wewnątrz palca, czyli w miejscu opuszki. Opisana metoda pozostawia jeszcze wiele możliwości zaimplementowania dodatkowego wnioskowania, na przykład dedukcji punktów więzadeł, identyfikacji palców oraz stopnia ich zgięcia w zależności od wahań różnic między ekstremami.

2.3.2. Analiza transformaty odległościowej

LeDung opisuje w swojej publikacji [79] algorytm wykrywania punktów charakterystycznych wykorzystując transformację odległościową oraz transformacje Hougha. Opisywany algorytm ma na celu wykrywanie dłoni w obrazach cyfrowych poprzez określenie środka regionu śródręcza oraz wyznaczenie pozycji wzniesionych palców. Z transformaty odległościowej wybierane są punkty charakterystyczne, a następnie z pomocą transformacji Hougha wykrywane są proste przechodzące przez zgrupowania wybranych punktów. Zgodnie z metodologią, odcinek reprezentujący palec znajdywany jest na wykrytej prostej, ograniczony skrajnymi pikselami danej grupy punktów.

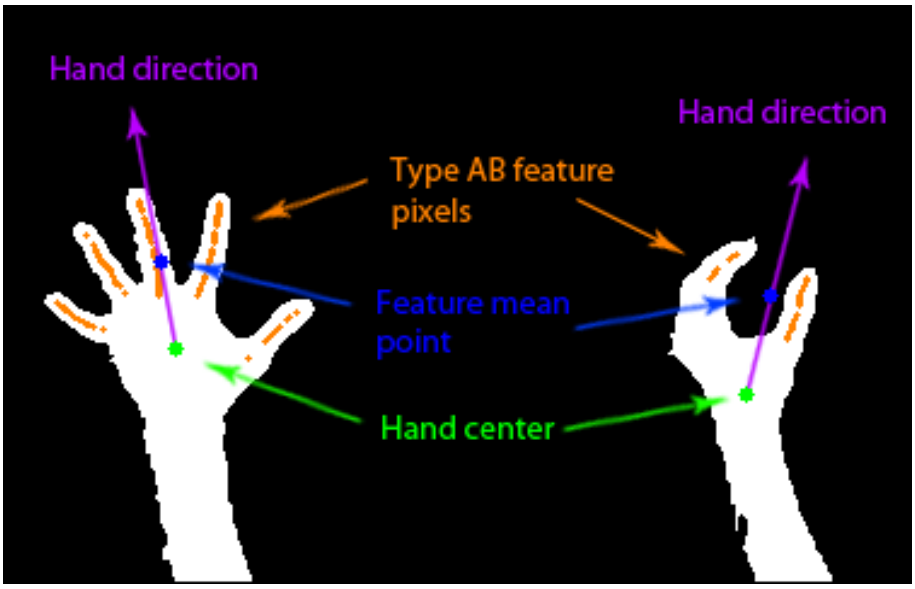
W celu porównania algorytmów wykrywania punktów charakterystycznych użyto implementacji Macieja Czupryny [80, 81]. W pracy opisuje on opracowany przez siebie program do interakcji człowiek-komputer działający na bazie analizy wyniku funkcji transformacji odległościowej, czyli transformaty odległościowej. Program służy do sterowania kursorem komputera za pomocą zdefiniowanych gestów. Obraz z kamery zamocowanej nad komputerem poddawany jest procesowi poszukiwania regionu skóry, następnie punkty charakterystyczne odnajdywane są na transformacie odległościowej. Ostatecznie, punkty te są analizowane w celu ustalenia pokazywanego gestu. Sam etap drugi, czyli algorytm poszukiwania punktów charakterystycznych, wygląda następująco:

Algorytm 1: Poszukiwanie punktów charakterystycznych w transformacie odległościowej

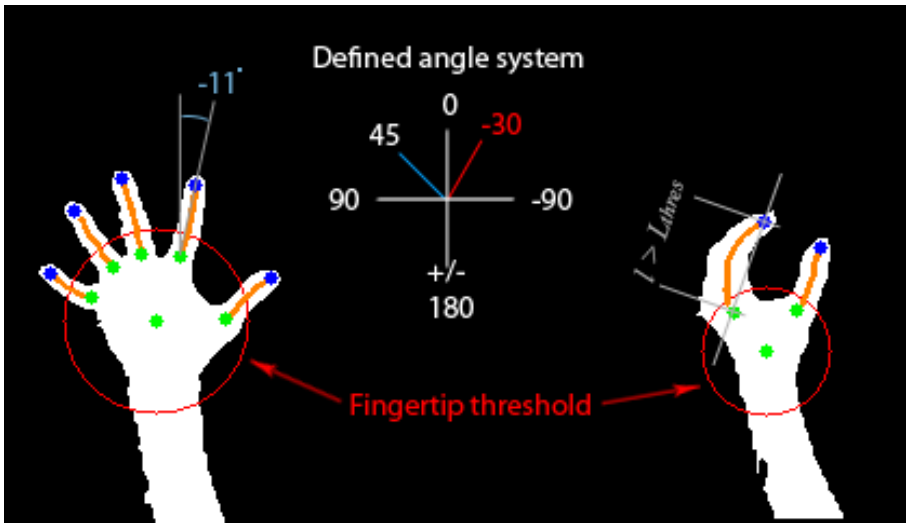
1. Wyznacz transformatę odległościową.
2. Znajdź współrzędne punktu maksimum oraz jego wartość. Punkt ten będzie środkiem śródręcza, a jego wartość odpowiadać będzie promieniowi koła wpisanego w kontur dłoni.
3. Wyznacz punkty lokalnych maksimów transformaty odległościowej. Będą to punkty typu A, które znajdują się w okolicach szkieletu obrazu. Za punkt typu A uznaje się punkt, który wokół swoich 8 sąsiadów ma wartość największą lub równą największej.
4. Zastosuj kryterium pozwalające na wyłonienie z punktów typu A punkty typu B, które będą ulokowane wzdłuż palców. Są to takie punkty, których wartość transformaty odległościowej w ich współrzędnych mieści się w zadeklarowanym przedziale.
5. Określ kierunek dłoni na podstawie momentów zbioru punktów typu B.
6. Wyznacz punkty typu C, które będą tworzyć krzywą, łączącą czubek palca z nasadą palca. Punkty typu C wyznacza się w podobny sposób jak punkty typu AB, jednak wystarczy, że posiadają wartość maksymalną wśród 7 z 8 swoich sąsiadów.

Dodatkowo, w algorytmie zdefiniowanych jest wiele kroków usprawniających sposób detekcji, pozytywnie wpływających na jakość wyniku końcowego. Dzięki uniwersalnej implementacji swojego algorytmu oraz nawiązaniu wzajemnej współpracy, algorytm ten mógł zostać przeniesiony do opisywanego środowiska testowego bez ingerencji w kod źródłowy.

Opisana metoda wykazuje dużą skuteczność wykrywania punktów charakterystycznych, w przypadku, gdy palce są oddzielone i wyprostowane. Algorytm ten w dokładny sposób wyznacza czubek i nasadę palca, jednocześnie pozwalając na dedukcję pozostałych punktów więzadeł. Takie rozwiązanie sprawdza się w odniesieniu do zdefiniowanego zestawu gestów, które zostały użyte na potrzeby sterowania kursorem komputera.



Rys. 8 Przykład wyznaczenia punktów typu AB (pomarańczowy), oraz środka dłoni (zielony) i orientacji dłoni (fioletowy). Rysunek pobrany z publikacji autora [80]



Rys. 9 Ostateczne wyznaczenie punktów charakterystycznych, nasady oraz czubków palców na podstawie punktów typu C. Rysunek pobrany z publikacji autora [80]

Niestety, obraz transformaty odległościowej nie pozwala na rozróżnienie palców w przypadku, gdy są one złączone lub stykają się w niektórych miejscach. Dlatego, na większości obrazów, mimo dużej liczby wykrytych punktów, są one wskazywane niedokładnie. W przypadku, gdy wszystkie palce są złączone, program „widzi" to jako jeden gruby palec.

2.3.3. Dopasowanie wzorca koła do maski dłoni

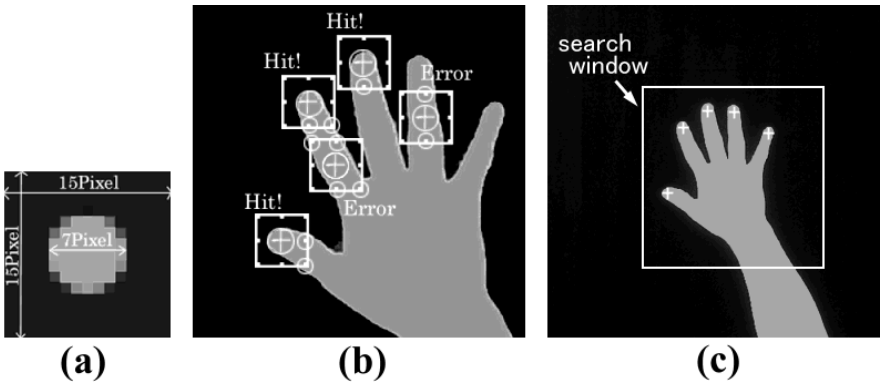
Metoda dopasowania wzorca, która opisana jest w rozdziale 3.2.3, ma wiele zastosowań w zależności od obrazu wejściowego oraz dopasowywanego wzorca. Y. Sato w swojej publikacji [78] opisywał możliwości zastosowania metody dopasowania wzorca koła do maski dłoni. Dokładny algorytm zawierał się w następujących krokach:

Algorytm 2: Wykrywanie punktów charakterystycznych za pomocą dopasowania wzorca

1. Przeprowadź operacje dopasowania wzorca koła do maski dłoni.
2. Przeszukaj wynik dopasowania wzorca w poszukiwaniu lokalnych maksimów odnajdując 20 punktów.
3. Przeprowadź sprawdzenie wykrytych punktów w celu filtracji fałszywych pozytywów.

Pierwszy krok algorytmu polega na przeprowadzeniu metody dopasowania wzorca. Autor nałożył serie ograniczeń odnośnie rozmiarów wzorca oraz samego obrazu. Jako że opisywana metoda stosowana jest do poszukiwania dłoni w znanej odległości od kamery, autor wykazał, że dłonie różnych użytkowników mają pomijalny wpływ na jakość detekcji. Pozwoliło to na sformułowanie założenia, w którym zakłada się stały region poszukiwań o rozmiarach 90×90 pikseli oraz stały rozmiar wzorca jako obraz o wymiarach 15×15 pikseli z kołem o średnicy 7 pikseli. Liczby te dobrane zostały w taki sposób, aby średnica wzorca koła była jak najbardziej zbliżona do średniej grubości palców. Właściwość ta została wytłumaczona również przy okazji opisu procedury walidacyjnej w rozdziale 5. Po ustaleniu rozmiarów

obrazów wejściowych autor zdecydował się na algorytm poszukiwania wzorca z użyciem znormalizowanej funkcji wzajemnej korelacji. Dokładny sposób działania funkcji opisano w rozdziale 3.2.3.



Rys. 10 Przykład działania algorytmu. a) wzorzec, b) maska z wykrytymi punktami, c) określenie regionu poszukiwań. Rysunek pobrany z publikacji autora [78]

Posiadając macierz prawdopodobieństwa dopasowania wzorca można przystąpić do etapu drugiego. Na tym etapie należy znaleźć w obrazie maksimum globalne i zapisać jego współrzędne do wektora z punktami zawierającymi potencjalne punkty charakterystyczne. Po wykryciu takiego punktu na obraz nanoszony jest obiekt wymazujący część informacji w otoczeniu wykrytego maksimum. Dzięki usunięciu wykrytego maksimum, w zmodyfikowanym obrazie będzie można poszukać nowego maksimum globalnego, w innym punkcie będącym nowym punktem charakterystycznym. Wymazanie dodatkowo regionu w bliskiej odległości od wykrytego punktu zapobiega wykrywaniu punktów charakterystycznych zbyt blisko siebie.

Oczywiście nie wszystkie z wykrytych punktów są poprawnymi punktami charakterystycznymi dłoni. Niektóre z nich należy odrzucić. Po pierwsze, nowo wykryty punkt nie może znajdować się zbyt blisko innych, po drugie, punkt czubka palca nie może znajdować się wewnątrz dłoni lub wewnątrz palca. W tym celu, w przypadku gdy zostanie wykryty punkt, w jego współrzędnych nanoszone jest koło o średnicy równej średnicy wzorca. Koło wypełnione jest kolorem tła tak, aby zastępowało potencjalne maksima wartością 0 w miejscach obok nowo wykrytego punktu. Kolejny etap to filtracja fałszywych pozytywów, tak jak zaprezentowano na Rys. 10 b. Na obrzeżach wzorca nowo wykrytego punktu wyznaczane jest 8 punktów. Jeżeli wykryty punkt ma być czubkiem palca, wtedy żadna para punktów leżących naprzeciwko siebie nie może mieć wartości różnej niż 0. W przypadku gdyby dla wykrytego punktu można było odnaleźć 2 przeciwległe punkty na obrzeżach wzorca nie będące tłem, znaczyłoby to, że punkt został wykryty, na przykład, wewnątrz palca, a nie na jego czubku. Takie punkty są odrzucane.

Ostatecznie algorytm powinien zwrócić od 0 do 5 punktów charakterystycznych będących czubkami palców. Czasem jednak zdarza się, że punkty te są wykryte na granicy nadgarstka lub w miejscach zgięć, gdy palce są schowane i widoczna jest wystająca kostka. W takich przypadkach kryterium filtracji fałszywych pozytywów nie zawsze działa. Dodatkowo algorytm jest w dużym stopniu zależny od odpowiedniego dobrania rozmiarów wzorca. Są to

jednak przypadki rzadkie, a ogólna zasada działania algorytmu gwarantuje dużą dokładność wykrytych punktów, mimo iż czasem nie są one zbyt liczne. Dodatkowo dzięki swojej prostocie algorytm wykonywany jest szybko.

2.3.4. Samoorganizująca się mapa neuronowa

Innowacyjne podejście do tematu wykrywania punktów charakterystycznych zostało zaproponowane w publikacji opisującej algorytm polegający na dopasowaniu samoorganizującej się sieci neuronowej do maski dłoni i analizy położenia neuronów [74]. Nazwa algorytmu pochodzi od tłumaczenia tytułu artykułu brzmiącego: Self Growing and Self Organizing Neural Gas (SGONG).

Etap pierwszy opisuje wykrywanie regionu skóry zastosowane na przestrzeni barw YCbCr. Wartości progów filtrujących piksele zostały dobrane po przeanalizowaniu zbioru uczącego. Ostatni krok okrojony jest do identyfikacji wzniesionych palców na podstawie wyznaczenia reguł, które za pomocą analizy kąta nachylenia palców oraz ich wzajemnej odległości są w stanie rozróżniać palce. Najważniejszą częścią całej publikacji jest sposób wyznaczenia punktów charakterystycznych, czyli nasady oraz czubka palców.

Algorytm podzielony jest na kilka ważniejszych etapów. Są nimi wykrywanie nadgarstka oraz ustalenie regionu śródręcza, naniesienie na maskę dłoni mapy sieci neuronów, ustalenie kąta orientacji dłoni oraz identyfikacja palców.

Należy wspomnieć, że oryginalnie algorytm zakłada, iż dłoń skierowana jest ku górze, tak jak przedstawiono to na Rys. 11. W celu wyznaczenia punktów nadgarstka i regionu śródręcza, liczony jest profil obrazu oraz szukane są lokalne ekstrema, podobnie jak w metodzie opisanej w rozdziale 4.1. Globalne maksimum odpowiada środkowi dłoni, a pierwsze lokalne minimum poniżej maksimum odpowiada regionowi nadgarstka. Odległość między punktem nadgarstka a środkiem dłoni odkładana jest w górę, tak by stworzyć całkowity region śródręcza bez palców. Na Rys. 11 b widać szary odcinek wewnątrz dłoni. Odcinek ten zaczyna się w punkcie środkowym regionu śródręcza wykrytego jako globalne maksimum profilu, a kończy w punkcie środkowym nadgarstka, wykrytego jako pierwsze lokalne minimum na profilu poniżej środka dłoni. Widać również dwie linie poziome, które przecinają obraz. Dolna przechodzi przez punkt nadgarstka, a górna podniesiona jest w stosunku do punktu nadgarstka o dwukrotność odległości między punktem nadgarstka a punktem śródręcza. W ten sposób autor postanowił określać region śródręcza.

Najważniejszym etapem jest naniesienie rozprzestrzeniającej się i samoorganizującej się sieci neuronowej na maskę dłoni. Autor dokładnie opisał algorytm dopasowania mapy neuronów w 8 krokach. Przed uruchomieniem algorytmu należy wylosować dwa punkty będące dwoma startowymi neuronami *W*.

Algorytm 3: SGONG [74]

1. Wyzeruj wszystkie wartości błędów nagromadzonych *AE=N=0* oraz ustaw parametry sterujące możliwościami ruchowymi neuronów do wartości domyślnych. Wylosuj zbiór punktów należących do maski dłoni (5% z losowych punktów z maski) zapisując wszystkie punkty w ciągu *X*.
2. Dla kolejnych wylosowanych punktów z ciągu X dopasuj dwa najbliżej znajdujące się neurony *W1* i *W2*.
3. Wyznacz parametry błędów dopasowania dla obu neuronów

$$\begin{aligned} AE &= AE + \|X_k - W\| , \\ N_{W1} &= N_{W1} + 1 . \end{aligned} \tag{3}$$

4. Wyznacz lub zmodyfikuj parametry plastyczności neuronu, tak aby przy pierwszym ruchu mógł przemieścić się o duży dystans, a z czasem by jego możliwości ruchowe spadały

$$\begin{aligned} e1_{W1} &= e1_{max} + e1_{min} - e1_{min}\left(\frac{e1_{max}}{e1_{min}}\right)^{N_{W1}/N_{iddle}} , \\ r_{W1} &= r_{max} + 1 - r_{max}\left(\frac{1}{r_{max}}\right)^{N_{W1}/N_{iddle}} . \end{aligned} \tag{4}$$

5. Przemieść neurony bliżej punktu, do którego zostały dopasowane zgodnie z ich możliwościami ruchowymi

$$W_{W1} = W_{W1} + e1_{W1}(X_k - W_{W1}) . \tag{5}$$

6. Dodaj lub usuń sąsiedztwo. Dwa neurony odnawiają sąsiedztwo, jeżeli oba były wykryte jako para w punkcie 2. Dwa neurony tracą sąsiedztwo, jeżeli ich sąsiedztwo jest zbyt stare lub linia łącząca je przechodzi przez tło.
7. Wykonaj kroki 2-6 dla wszystkich elementów zbioru punktów. Ten krok kończy epokę. Jeżeli wszystkie neurony są w stanie ustalonym ($N_W = N_{iddle}$) zakończ algorytm.
8. Wykonaj kryterium odrzucania lub dodawania neuronów w okolicy neuronu z największym błędem dopasowania *AE*. Neuron zostaje usunięty, jeżeli od kilku epok nie został użyty. Neuron zostaje dodany w miejscu największego nagromadzenia błędu dopasowania z punktu 3.

Warto zauważyć podział na iteracje oraz tzw. epoki. Autor używa słowa „epoka” aby określić pętlę, która zawiera w sobie inną pętlę iteracyjną. Nazewnictwo takie można uargumentować strukturą algorytmu, która w punktach 2-7 realizuję pętlę iteracyjną, tak długo, aż wyznaczane punkty ustabilizują się. Zresetowanie parametrów oraz dodanie nowych nieprzypisanych punktów destabilizujących siatkę rozumiane jest jako początek nowej epoki.

W celu przygotowania potrzebnych danych do algorytmu należy wylosować zbiór zawierający 5% losowych pikseli z maski dłoni. Autor tłumaczy zasadność użycia dokładnie 5% danych argumentem wskazującym, że większa liczba danych negatywnie wpływa na czas nie zwiększając precyzji [74]. Jeżeli maska dłoni zawierać będzie 3000 pikseli oznaczających region skóry, 150 elementów zostanie wybranych. Algorytm zatem przeprowadzi 150 iteracji, czyli kroków 2-7, tyle razy ile epok, czyli kroków 1-8, jest wymaganych. W przypadku gdy te 150 iteracji nie sprawi, że neurony ustabilizują się, wykonywana jest kolejna epoka, neurony są dodawane lub usuwane, a parametry resetują się.

Powyższy opis został zaimplementowany, jednak na podstawie publikacji nie było możliwe ustalenie dokładnych wartości parametrów sterujących.

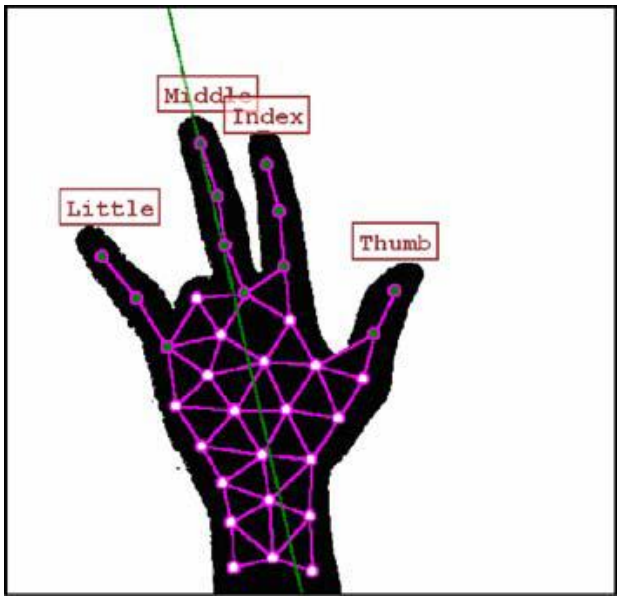
Mimo przedstawienia dokładnego opisu wyżej wymienionego algorytmu, który w publikacji zawiera wszelkie niezbędne wzory i nie pozostawia wątpliwości, autor postanowił nie publikować nastaw parametrów sterujących algorytmem. Po zaimplementowaniu algorytmu zgodnie z opisem autora, pojawił się problem jak poprawnie dopasować parametry aby algorytm działał dokładnie tak, jak w artykule. W tym celu stworzono listę parametrów sterujących oraz przeprowadzono serię testów dla różnych wartości parametrów. Wartości dobrane zostały w taki sposób, aby przede wszystkim algorytm dawał wizualnie podobne wyniki. Zbiór dobranych sterujących przedstawiony jest w Tab. 1.

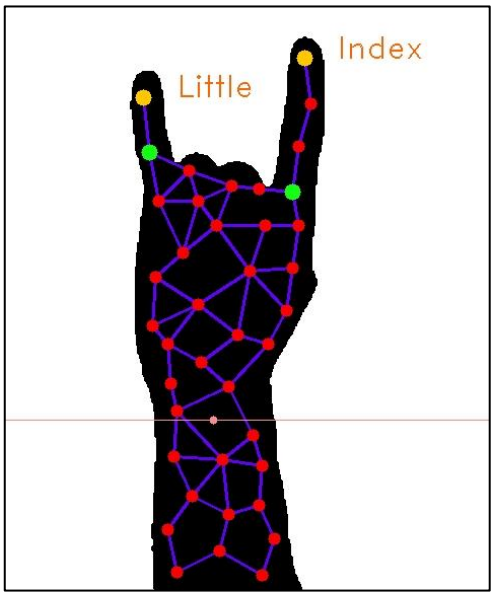
Tab. 1 Zbiór parametrów sterujących algorytmem SGONG wraz z ich opisem

Zmienna	Wartość	Opis
K	300	Kryterium stopu w razie wejścia w pętle nieskończoną.
N_{Idle}	50	Ile iteracji wskazujących dany neuron musi minąć, aby uznać go za neuron w stanie ustalonym.
N_{Size}	40	Maksymalna liczba neuronów.
d_{max}	110 [%]	Kryterium odrzutu sąsiedztwa. Jeżeli dwa neurony znajdują się bliżej niż wartość 0,6 lub dalej niż 1,1 razy średnia odległość między sąsiadami, wtedy dane sąsiedztwo zostaje zerwane.
d_{min}	60 [%]	
s_{old}	10	Jeżeli sąsiedztwo nie zostanie odnowione w przeciągu 10 epok, zostaje ono zerwane.
N_{old}	3	Jeżeli neuron nie zostanie użyty w przeciągu 3 epok, zostaje on skasowany.
$e1_{max}$	0,45	Parametry sterujące możliwościami ruchowymi neuronów. Nowo wykryty punkt przyciągnie przypisany do niego neuron o dystans równy 0,45 razy ich wzajemna odległość. Możliwości ruchowe będą się zmniejszać, osiągając wartość 0,025 po 30-tym przypisaniu punktu do danego neuronu.
$e1_{min}$	0,025	
r_{max}	30	

Na Rys. 11, po lewej stronie widać oryginalny obraz pobrany z publikacji autora algorytmu SGONG, a po prawej wynik zaimplementowanego algorytmu działający z wykorzystaniem parametrów sterujących z Tab. 1. Niestety, ustalanie wartości parametrów przebiegało w sposób wizualny, to znaczy celem było uzyskanie obrazu wyglądającego tak samo jak obraz z publikacji autora. Takie podejście było wystarczające, ponieważ błędne dopasowanie

parametrów kończyło się tworzeniem sieci neuronów o bardzo nieregularnych kształtach. Dodatkowo, błędu, wynikającego ze złego dopasowania parametrów, nie dało się w żaden sposób zmierzyć.

a) 

b) 

Rys. 11 Przykład wyniku końcowego algorytmu SGONG. a) Obraz pobrany z publikacji autora [74]. b) Obraz uzyskany po zaimplementowaniu algorytmu od podstaw.

Kolejnym krokiem po ustabilizowaniu się sieci neuronów było określenie pozycji palców oraz ich identyfikacji. Każdy neuron, który ma tylko jednego sąsiada, traktowany jest jako czubek palca. Region palca wyznaczają kolejne neurony aż do takiego neuronu, który ma więcej niż 2 sąsiadów. Dzięki temu ustalany jest punkt czubka oraz nasady palca.

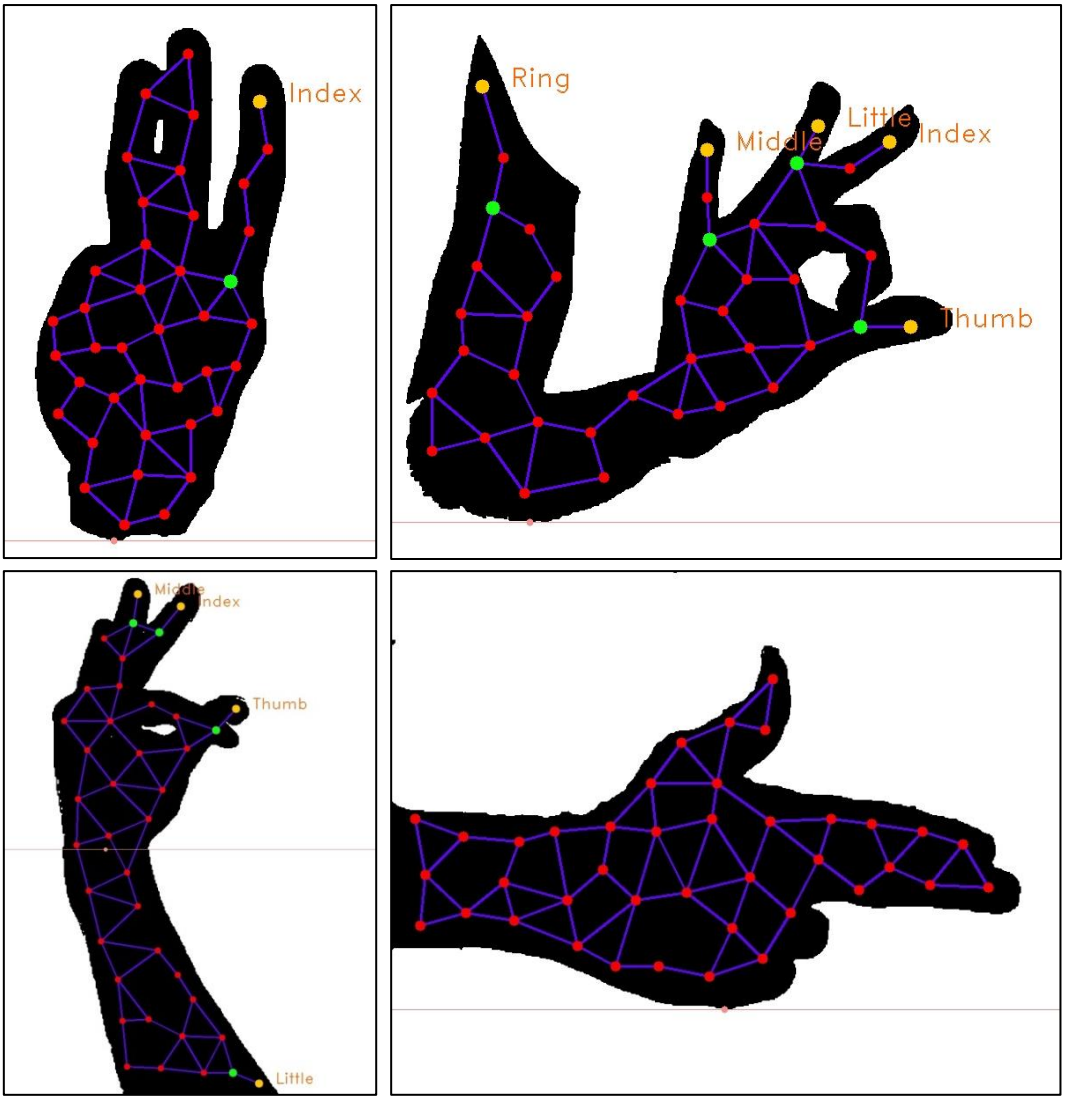
Ostatecznie, autor opisywał również metodę identyfikacji palców na postawie wzajemnych kątów i odległości między punktami charakterystycznymi. Najpierw wyznaczany był kąt orientacji dłoni składający się z kąta wyliczonego dwoma metodami. Pierwszy kąt określało nachylenie prostej przechodzącej przez punkty środkowe śródręcza oraz nadgarstka, a drugi nachylenie prostej przechodzącej przez dwa skrajnie lewe neurony, znajdujące się w regionie śródręcza najbliżej linii oddzielających region śródręcza. Odcinki kątów nachylenia dłoni widać na Rys. 11 b, gdzie szare odcinki pokazują obie składowe ostatecznego kąta nachylenia, oraz na Rys. 11 a, gdzie widać zieloną prostą reprezentującą ostateczny kąt nachylenia dłoni, przechodzącą przez środek śródręcza. Po wyznaczeniu prostych określane zostały trzy miary:

- RC: Różnica między kątem nachylenia dłoni oraz nachylenia prostej łączącej nasadę palca (Root) ze środkiem dłoni (Center);
- TC: Różnica między kątem nachylenia dłoni oraz nachylenia prostej łączącej czubek palca (Tip) ze środkiem dłoni (Center);
- D: Odległość (Distance) pomiędzy nasadą palców a linią wyznaczającą kąt nachylenia dłoni.

Miary te zostały wyznaczone za pomocą zbioru uczącego, tak, aby nowo wykryte punkty mogły być identyfikowane i klasyfikowane za pomocą tych wartości.

Warto jednak wspomnieć o głównych wadach i zaletach samego procesu wykrywania i lokalizacji punktów charakterystycznych. Algorytm wykrywa nie tylko punkty czubka palców, ale również nasady palców. Niestety, część zdjęć nie jest analizowana z powodu założeń odnośnie pionowej orientacji dłoni i poprawnego kadrowania. Restrykcyjne założenia sprawiają, że większość zdjęć z bazy posiada błędnie wykryty region śródręcza na pierwszym etapie algorytmu. Często zdarzają się obrazy, dla których nadgarstek zostaje wykryty w nieodpowiednim punkcie lub nie zostaje wykryty wcale, ponieważ globalne maksimum profilu znajduje się w regionie łokcia. Ręce, które ustawione są poziomo na obrazie, również mają źle wykryty region nadgarstka.

Jednak sam region nadgarstka oraz orientacja dłoni powodują jedynie błędną identyfikację. Błędy związane z wykrywaniem i lokalizacją punktów charakterystycznych spowodowane są najczęściej tym, że dla zwartych regionów, jak na przykład zaciśnięta pięść lub połączone palce, neurony mają dużo miejsca oraz wielu sąsiadów w danym regionie. Prowadzi to do braku pojedynczego neuronu, który mógłby zapoczątkować wykrycie palca. Dodatkowo, ostro zakończony region rękawa sprawia, że palce wykrywane są w miejscu łokcia. Kilka przykładów opisanych powyżej problemów można odnaleźć na Rys. 12.



Rys. 12 Przykłady częstych błędów algorytmu SGONG

3. Wybrane metody sztucznej inteligencji i przetwarzania obrazów

Metody sztucznej inteligencji oraz przetwarzania obrazów posiadają pewną część wspólną. W książce autorstwa S. Russell oraz P. Norvig [82] dotyczącej metod sztucznej inteligencji, w rozdziale poświęconym percepcji, autorzy wyróżniają przetwarzanie obrazów jako narzędzie służące agentowi inteligentnemu do pozyskiwania informacji o otoczeniu. Podobnie jak zmysły słuchu, dotyku oraz wzroku u człowieka, odpowiednie czujniki akustyczne, sensoryczne oraz wizyjne mogą posłużyć agentowi inteligentnemu do zbierania informacji.

Metody analizy obrazów służą do pozyskiwania oraz przetwarzania danych wizyjnych. Celem tych metod jest między innymi uzyskanie jak największej i jak najdokładniejszej reprezentacji otoczenia. W przypadku wykrywania punktów charakterystycznych, celem algorytmu przetwarzania obrazu jest utworzenie zbioru punktów charakterystycznych, które następnie poddaje się analizie.

Analiza pozyskanych danych wizyjnych oraz danych uzyskanych z algorytmów przetwarzania obrazu mieści się w dziedzinie sztucznej inteligencji. Celem algorytmów sztucznej inteligencji jest między innymi selekcja oraz klasyfikacja informacji w taki sposób, aby jak najdokładniej przedstawić informacje o otoczeniu. W przypadku wykrywania punktów charakterystycznych, algorytmy sztucznej inteligencji służą do odrzucania punktów błędnych, klasyfikacji punktów wykrytych lub wnioskowania o punktach dodatkowych.

3.1. Metody sztucznej inteligencji

Definicja sztucznej inteligencji nie jest jednoznaczna, a różne jej definicje zostały przedstawione w książce [82]. Jednak wszystkie opisy wskazują, że sztuczna inteligencja jest to myślenie lub zachowywanie się w sposób racjonalny lub naśladujący człowieka. Ważną cechą algorytmów sztucznej inteligencji jest umiejętność podejmowanie decyzji. Oznacza to, że program komputerowy będzie w stanie rozróżniać pozyskiwane informacje zgodnie z zaimplementowanymi zasadami.

Samo wykrycie cech charakterystycznych na obrazie cyfrowym należy do dziedziny przetwarzania obrazu. Wybór odpowiednich cech, klasyfikacja wykrytych punktów oraz wnioskowanie o położeniu dodatkowych punktów należą do dziedziny sztucznej inteligencji. Podstawowymi metodami wykorzystywanymi w pracy są drzewa decyzyjne oraz wnioskowanie.

3.1.1. Wnioskowanie

Wnioskowanie jest to proces rozumowania pozwalający na uznanie za prawdziwe zdań na podstawie prawdziwości innych zdań [83]. Decyzja podejmowana jest za pomocą bazy wiedzy, czyli zbioru definicji, faktów pojęć i relacji między nimi. Oprogramowanie wnioskujące ma na celu odnalezienie rozwiązania poprzez manipulowanie bazą wiedzy.

Wyróżnia się wnioskowanie progresywne, regresywne oraz mieszane. Wnioskowanie progresywne, nazywane wnioskowaniem od przodu, polega na utworzeniu nowego faktu za pomocą dostępnych reguł. Oznacza to, że system taki posiada zdefiniowane zbiory reguł, a nowo powstały przypadek określany jest zgodnie z tymi regułami.

W celu wyjaśnienia wnioskowania progresywnego przedstawiono przykład zawierający następujące reguły:

- Punkt nasady palca znajduje się w regionie śródręcza
- Punkt nasady palca nie może znajdować się zbyt blisko punktu nadgarstka

Jeżeli za pomocą algorytmów przetwarzania obrazów wykryto nowy punkt, którego pierwszy atrybut określa przynależność do regionu śródręcza, a drugi określa bliską odległość od nadgarstka, wtedy zgodnie z określonymi regułami, punkt nie może być punktem nasady palca.

Wnioskowanie regresywne, nazywane wnioskowaniem wstecz, polega na określeniu nowej reguły decyzyjnej, za pomocą zaobserwowanych faktów. W przypadku wykrycia wielu faktów, które posiadają podobne atrybuty, słuszna jest ich klasyfikacja tworząca nową regułę decyzyjną. Analiza wyników działania algorytmu oraz wniosków implementacyjnych może przyczynić się do powstania nowej reguły, za pomocą zaobserwowanych faktów. Przykładem może być sformułowanie reguły decyzyjnej odrzucającej punkty znajdujące się zbyt blisko konturu, po zaobserwowaniu, że punkty te, na ogół, posiadają wysoki błąd dopasowania.

Wnioskowanie mieszane łączy wnioskowanie progresywne z regresywnym, łącząc wnioskowanie od przodu z wnioskowaniem od końca. Przykładem jest wykrywanie nowego punktu za pomocą określonych reguł oraz za pomocą wzajemnego położenia punktów już wykrytych. Zastosowane jest to w algorytmie autorskim, w przypadku określenia położenia punktu nasady palca względem czubka palca.

3.1.2. Drzewo decyzyjne

Drzewo decyzyjne należy do kategorii uczenia indukcyjnego. Składa się ono z węzłów decyzyjnych, węzłów stanu natury oraz gałęzi. Drzewo decyzyjne jest to graf, którego wierzchołek początkowy przedstawia wybrany atrybut, natomiast krawędzie określają reguły decyzyjne zależne od wartości tego atrybutu. Gałęzie prowadzą do węzłów, które przedstawione są z użyciem kolejnych atrybutów. Koniec grafu przedstawiają poszczególne klasy. Decyzja o klasyfikacji danego obiektu podejmowana jest za pomocą wartości jego atrybutów [84].

Drzewo decyzyjne przedstawia usystematyzowany zbiór reguł decyzyjnych, w którym każda następna decyzja zależna jest od poprzedniej. Za pomocą drzewa decyzyjnego można przeprowadzić klasyfikację wykrytych obiektów za pomocą ich atrybutów oraz wcześniej utworzonych reguł decyzyjnych.

W przypadku algorytmów wykrywania punktów charakterystycznych, drzewo decyzyjne używane jest w celu odrzucenia punktów fałszywych lub określenia przynależności wykrytego punktu do danej grupy.

3.2. Metody analizy obrazu

Algorytmy wykrywania punktów charakterystycznych, stanowiące analizę obrazu cyfrowego, w większości przypadków wykorzystują elementy cyfrowego przetwarzania obrazów. Dodatkowo, ich specyfika wywodzi się głównie z zastosowania danej metody lub zbioru metod. Jednak sama istota danej metody nie ma większego znaczenia, w przeciwieństwie do jej efektu końcowego. Oznacza to, że celem badawczym algorytmu z użyciem transformaty odległościowej nie jest sposób implementacji tej transformacji, lecz możliwości detekcji lokalnych cech charakterystycznych, jakie ta metoda daje.

Program komputerowy zawierający opisywane algorytmy stworzony został w oparciu o bibliotekę OpenCV [85], gdzie większość znanych metod przetwarzania obrazu została zaimplementowana w optymalny sposób.

Ten rozdział zawiera opis wybranych, ważniejszych metod przetwarzania obrazu, które są kluczowe dla opisywanych, w dalszych rozdziałach, algorytmów wykrywania punktów charakterystycznych.

3.2.1. Metody segmentacji obszaru występowania skóry

Większość algorytmów analizy gestów przedstawiona jest jako trójetapowy schemat. Pierwszym etapem jest segmentacja regionu zainteresowania, czyli dłoni. Kolejne etapy to ekstrakcja cech z obrazu oraz wnioskowanie o wyniku. Przedmiotem niniejszej rozprawy doktorskiej jest etap ekstrakcji cech, czyli etap drugi. Niniejszy rozdział opisuje etap pierwszy ponieważ jego wynik bezpośrednio wpływa na kolejne etapy algorytmu analizy gestów.

Aby skutecznie przeprowadzić detekcję punktów charakterystycznych należy najpierw wyłonić z obrazu obszar ręki. Do tego celu wykorzystać można między innymi metody detekcji skóry. Dokładny opis algorytmów wykrywania skóry, jak i ich porównanie, można odnaleźć w [57, 58].

Informacją wyjściową z algorytmów wykrywania skóry jest maska dłoni lub ręki. Maska to jednokanałowy, jednobitowy obraz, którego piksele przyjmują wartości 1 lub 0. Dany piksel z maski przenosi informacje o tym, czy piksel z obrazu o tych samych współrzędnych należy do regionu ręki czy tła.

a) b) c)

Rys. 13 a) Obraz wejściowy reprezentujący rękę oraz tło, b) maska dłoni wskazana przez eksperta, c) mapa prawdopodobieństwa występowania regionu skóry.

Rys. 13 a przedstawia przykład obrazu, dla którego wyznaczono maski występowania regionu skóry. Warto zauważyć, że Rys. 13 b przedstawia maskę jednobitową z wartościami 0 lub 1, natomiast Rys. 13 c przedstawia maskę 8-bitową, w której informacja o regionie skóry

należy do zbioru rozmytego. Oznaczenie 0 lub 1 jako koloru czarnego lub białego nie jest jednoznaczne i różni się pomiędzy wieloma publikacjami. Na potrzeby prezentacji wyników w niniejszej pracy przyjęto standard, w którym kolor biały oznacza tło, a kolor czarny oznacza region występowania skóry. Wynika to z potrzeby przystosowania obrazu do druku, w którym kolorem tła, czyli kartki, jest kolor biały.

3.2.2. Transformacja odległościowa

Po etapie uzyskania maski dłoni, czyli po ustaleniu regionu występowania skóry w obrazie, większość operacji realizowanych jest na masce, a nie na obrazie barwnym. Tak dzieje się, na przykład, podczas stosowania transformacji odległościowej, którą w najprostszej postaci stosuje się właśnie na jednobitowym obrazie.

Wynikiem transformacji odległościowej dla maski z zaznaczonym obszarem dłoni oraz tła jest macierz transformaty. Jest ona przekształceniem morfologicznym, które w obrazie wyjściowym przyporządkowuje odległość każdego punktu o wartości 1 do najbliższego punktu o wartości 0 w obrazie wejściowym. Koszty przejścia funkcji transformacji odległościowej określa się wieloma metrykami. Przykłady popularnych metryk przedstawiono w Tab. 2.

Tab. 2 Koszty funkcji transformacji odległościowej

Metryka	Piksele sąsia-dujące przylegle	Piksele sąsiadujące po przekątnej
Czebyszewa	1	1
Manhattan	1	2
Euklidesowa	0,95509	1,36930

Metryka Czebyszewa wzorowana jest na ruchach, które pokonać może figura króla na szachownicy. Przejście, zarówno, do piksela sąsiadującego przylegle, jak i po przekątnej kosztuje 1 ruch. Metryka Manhattan wzorowana jest na planie miasta i trasie poruszającej się po niej taksówce. Aby przedostać się na róg przeciwległego budynku, należy przemieścić się wzdłuż dwóch jego boków. Dlatego koszt przejścia do sąsiada po przekątnej opisany jest kosztem równym 2. Metryka euklidesowa, wyznaczana z normą L^2, przyjmuje wartości zoptymalizowane, zgodnie z opisem przedstawionym w [86].

Liczenie transformaty odległościowej przebiega w następujący sposób. Wszystkie piksele tła umieszczane są w samoorganizującej się kolejce. Jako że każdy z nich ma wartość 0, w początkowej fazie ich kolejność nie ma znaczenia. Zaczynając od najmniejszych wartości wszystkim pikselom sąsiadującym, przypisywane są wartości piksela wyjściowego plus wartość zgodnie z Tab. 2, chyba że piksel jest pikselem tła bądź we wcześniejszym kroku przypisano mu niższą wartość. Nowo oznaczony piksel zapisywany jest w kolejce na odpowiednim miejscu. Dzięki samoorganizującej się strukturze kolejki, najpierw analizowane są piksele tła, a następnie te, które mają najniższą wartość, wykluczając pętle nieskończone. Przykład działania transformacji odległościowej typu L2 przedstawiony jest na Rys. 14.

0	0	0	0	0	0	0
0	0	0	1	1	1	0
0	1	0	1	1	1	1
0	1	1	1	1	1	1
0	1	1	1	1	1	1
0	1	1	1	1	1	1
0	0	0	0	1	1	1

0	0	0	0	0	0	0
0	0	0	0,95	0,95	0,95	0
0	0,95	0	0,95	1,9	1,37	0,95
0	0,95	0,95	1,37	2,32	2,32	1,9
0	0,95	1,9	1,9	2,32	2,74	2,85
0	0,95	0,95	0,95	1,37	2,32	3,27
0	0	0	0	0,95	1,9	2,85

Rys. 14 Przykład działania transformacji odległościowej typu L2. Po lewej fragment jednobitowej maski. Po prawej transformata odległościowa w postaci macierzy liczb rzeczywistych, w którym maksymalna wartość reprezentowana jest jako kolor biały.

Celem transformacji odległościowej jest, między innymi, określenie szerokości obiektu. Lokalna szerokość obiektu w danym punkcie rozumiana jest jako najkrótsza odległość łącząca przeciwległe punkty konturu przechodząca przez wskazany punkt. Analizując maksima lokalne transformaty odległościowej dłoni, można określić, jak gruby jest palec lub jak duży jest region śródręcza, oczywiście w skali pikseli. Dokładny opis metody wykorzystującej transformatę odległościową można odnaleźć w rozdziale 2.3.2.

3.2.3. Dopasowanie wzorców

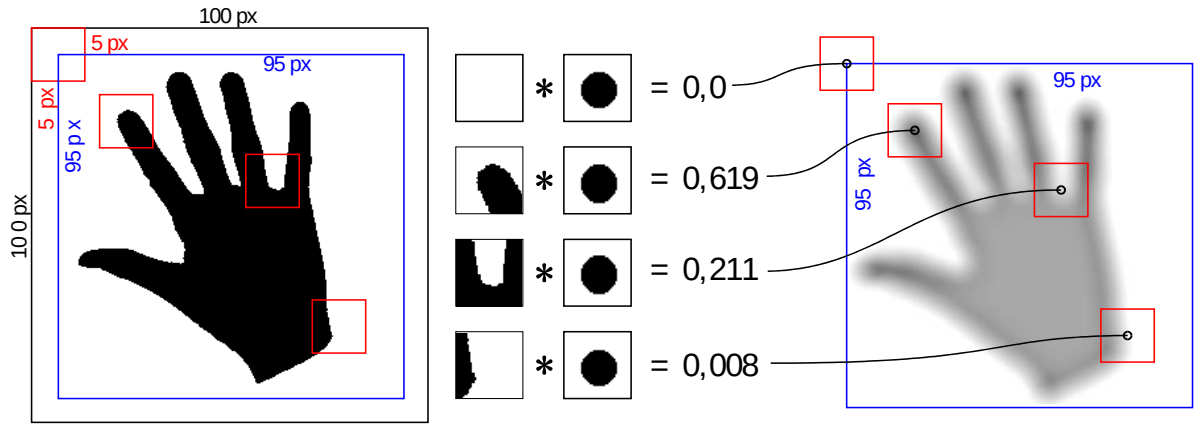
Szeroko rozumiana metoda dopasowania wzorca lub poszukiwania wzorca przyjmuje jako dane wejściowe obraz źródłowy oraz obraz przedstawiający wzorzec, który będzie poszukiwany. Wynik funkcji splotu dwóch obrazów przedstawiony jest jako macierz podobieństwa w skali procentowej lub w postaci liczb rzeczywistych, przyjmująca wartości 0–1. Macierz prawdopodobieństwa może zostać zwizualizowana w postaci obrazu tworząc mapę prawdopodobieństwa. Przykładami metody dopasowania wzorca (ang. template matching) mogą być poszukiwanie i identyfikacja znaków drogowych [87], analiza obrazu na potrzeby inżynierii biomedycznej [88] lub również opisywany w rozdziale 2.3.3 algorytm poszukiwania punktów charakterystycznych na obrazach dłoni [78].

Obraz źródłowy *Z(x,y)* można przedstawić jako funkcje dwóch zmiennych, w których dla danych współrzędnych x_i, y_j funkcja przyjmuje różne wartości przedstawiane jako piksele. Podobnie obraz wzorcowy *W(x',y')*, jest funkcją dwóch zmiennych x'_i, y'_j. Ważne jest, aby oba obrazy były w tej samej przestrzeni barw oraz aby rozmiar wzorca był mniejszy niż rozmiar obrazu źródłowego, czyli $x > x' \wedge y > y'$. W przypadku, gdy obrazy są w innej przestrzeni barw, pierwszym krokiem algorytmu jest konwersja do wspólnej przestrzeni barw. Najczęściej sprowadza się obrazy do skali szarości. W trakcie metody poszukiwania wzorca, wzorzec nakładany jest na kolejne dostępne punkty w obrazie źródłowym. W każdym punkcie liczony jest splot wzorca oraz obszaru na obrazie źródłowym pokrytym przez wzorzec.

Macierz prawdopodobieństwa, która jest splotem, czyli wynikiem operacji dopasowania wzorca, jest to macierz o szerokości i wysokości równej różnicy szerokości i wysokości obrazu źródłowego i obrazu wzorca. Powód zmiany rozmiarów obrazów zostanie wytłumaczony na podstawie następującego przykładu.

Zakładamy obraz źródłowy o wymiarach 100×100 pikseli oraz wzorzec o rozmiarach 5×5. Dopasowanie wzorca w miejscu *(0,0)-(5,5)* zapisze wynik do macierzy końcowej w pozycji *(3,3)*. Tak samo dopasowanie skrajnego wzorca w pozycji *(95,95)-(100,100)* również zapisze wynik w pozycji *(97,97)*. Z tego powodu nie ma możliwości uzyskania wyniku dopasowania wzorca w skrajnych krawędziach obrazu, jak na przykład punkty: *(0,1), (99,99), (40,2)*. Dlatego obraz docelowy będzie miał wymiary *95×95*. Należy o tym pamiętać lokalizując punkty lokalnych maksimów z odwzorowaniem ich na obraz źródłowy.

Dane, które zostaną zapisane do macierzy prawdopodobieństwa, czyli wyniku dopasowania wzorca, zależą od wybranej funkcji dopasowania wzorca. Na ogół będą to liczby rzeczywiste, których wartość odpowiadać będzie prawdopodobieństwu występowania wzorca w danym miejscu. Rys. 15 przedstawia sposób działania algorytmu dla wybranych 3 punktów. Wybrane punkty sprawdzane są ze wzorcem zgodnie z wybraną funkcją, a następnie wynik tego dopasowania przenoszony jest na macierz prawdopodobieństwa. Należy zwrócić uwagę na to, że liczby wskazane jako wynik mają tylko charakter poglądowy i nie są to prawdziwe wartości.



Rys. 15 Przykład działania algorytmu dopasowania wzorca koła o wymiarach 5×5 do maski dłoni o wymiarach 100×100. Pokazano wyniki splotu dla przykładowych 4 punktów. Obraz docelowy o wymiarach 95×95 przedstawia wyniki splotu każdego punktu.

Najczęstszymi metrykami liczenia prawdopodobieństwa spotu przedstawionego jako $P = Z * W$, są między innymi: suma kwadratów (ang. sum of squares), korelacja wzajemna (ang. cross-corelation), współczynnik korelacji (ang. correlation coefficient) oraz ich normowane odmiany. Najwydajniejszą metodą poszukiwania wzorca jest znormalizowana metoda wzajemnej korelacji [78]. Dokładna wartość nowego piksela w macierzy wyniku będzie zależała od wartości funkcji opisanej w równaniu:

$$P(x,y) = \frac{\sum_{x',y'}(\mathrm{W}(x',y') \cdot Z(x+x',y+y')}{\sqrt{\sum_{x',y'} \mathrm{W}(x',y')^2 \cdot \sum_{x',y'} Z(x+x',y+y')^2}}. \qquad (6)$$

Równanie (6) przedstawia funkcje normowanej wzajemnej korelacji dla macierzy prawdopodobieństwa (P), w zależności od obrazu źródłowego (Z) oraz wzorca (W). Podczas

badań algorytmów bazujących na metodzie dopasowania wzorca stosowana jest zaimplementowana w bibliotece OpenCV funkcja matchTemplate.

Zastosowanie metody poszukiwania czarno-białego wzorca na czarno-białej masce może być mylone z przekształceniami morfologicznymi lub filtracją. Gdyby na obrazie maski przeprowadzona została operacja morfologiczna o jądrze przypominającym wzorzec koła, wtedy w obrazie wyjściowym piksel o wartości „1" zostałby zapisany jedynie w punkcie, w którym dokładnie taki sam obiekt występował. Takie podejście wskazałoby jedynie miejsce idealnego dopasowania, a jak łatwo można zauważyć, takie miejsce w przypadku obrazów dłoni nie występuje prawie nigdy. Niestety, zarówno funkcje morfologiczne jak i filtracja dają wyniki w dziedzinie klasycznej logiki dwuwartościowej, dlatego tylko miejsca idealnego dopasowania zostałyby odnalezione. Przedstawione metody dopasowania wzorca działają na bazie prawdopodobieństwa dopasowania, dając wynik w przestrzeni logiki rozmytej. Na takiej macierzy można przeprowadzić operacje poszukiwania lokalnych maksimów, czyli regionów najlepszego (nie idealnego) dopasowania.

3.2.4. Obraz kierunkowy

Obraz maski dłoni przekazuje jedynie informacje o tym gdzie dłoń się znajduje, oraz jaki jest jej kształt. Z kształtu można wyznaczyć regiony charakterystyczne, takie jak nadgarstek, region śródręcza, wystające palce. Sam obraz maski jednak nie przenosi informacji o tym, co znajduje się wewnątrz niej.

Większość algorytmów bazuje na analizie konturu pozyskanego z obwodu maski dłoni. Aby zwiększyć liczbę wykrytych punktów charakterystycznych należy zatem zaproponować rozwiązanie odnajdujące kontury wewnętrzne, które wyszczególnią między innymi zgięte palce schowane wewnątrz maski. Jedna z metod pozyskiwania takiego rodzaju informacji, wykorzystująca obraz kierunkowy, opisana została w artykule [89].

Obrazem wejściowym jest obraz barwny, przekształcony na obraz w skali szarości. W trakcie przekształceń, dla grupy 9 pikseli, liczony jest kierunek stycznej do gradientu, wartość błędu dopasowania oraz waga dopasowania. Wynikiem jest macierz zawierająca wektory kierunków. Obraz kierunkowy jest wizualizacją tej macierzy, czyli jest to obraz binarny, w którym dla wybranych pikseli zrenderowane zostały odcinki nachylone pod kątek określonym przez gradient lokalny.

Pierwszym etapem jest wyznaczenie operatora Sobela dla każdej grupy pikseli o wymiarach 3×3, zgodnie z równaniem (7), przedstawiającym operator Sobela dla macierzy I o wymiarach 3×3. Dla każdego piksela w obrazie wyznaczany jest wektor gradientu *($g_{n,m}$),*

$$g_{m,n} = \begin{bmatrix} I_{m+1,n+1} + I_{m+1,n-1} - I_{m-1,n+1} - I_{m-1,n-1} + 2I_{m+1,n} - 2I_{m-1,n} \\ I_{m+1,n-1} + I_{m-1,n-1} - I_{m-1,n+1} - I_{m+1,n+1} + 2I_{m,n-1} - 2I_{m,n+1} \end{bmatrix}. \quad (7)$$

Następnie dla co trzeciego piksela w co trzecim rzędzie wyznaczana jest styczna do gradientu *($u_{n,m}$)* w tym punkcie, bazując na wartości gradientu tego piksela oraz 24 sąsiadów,

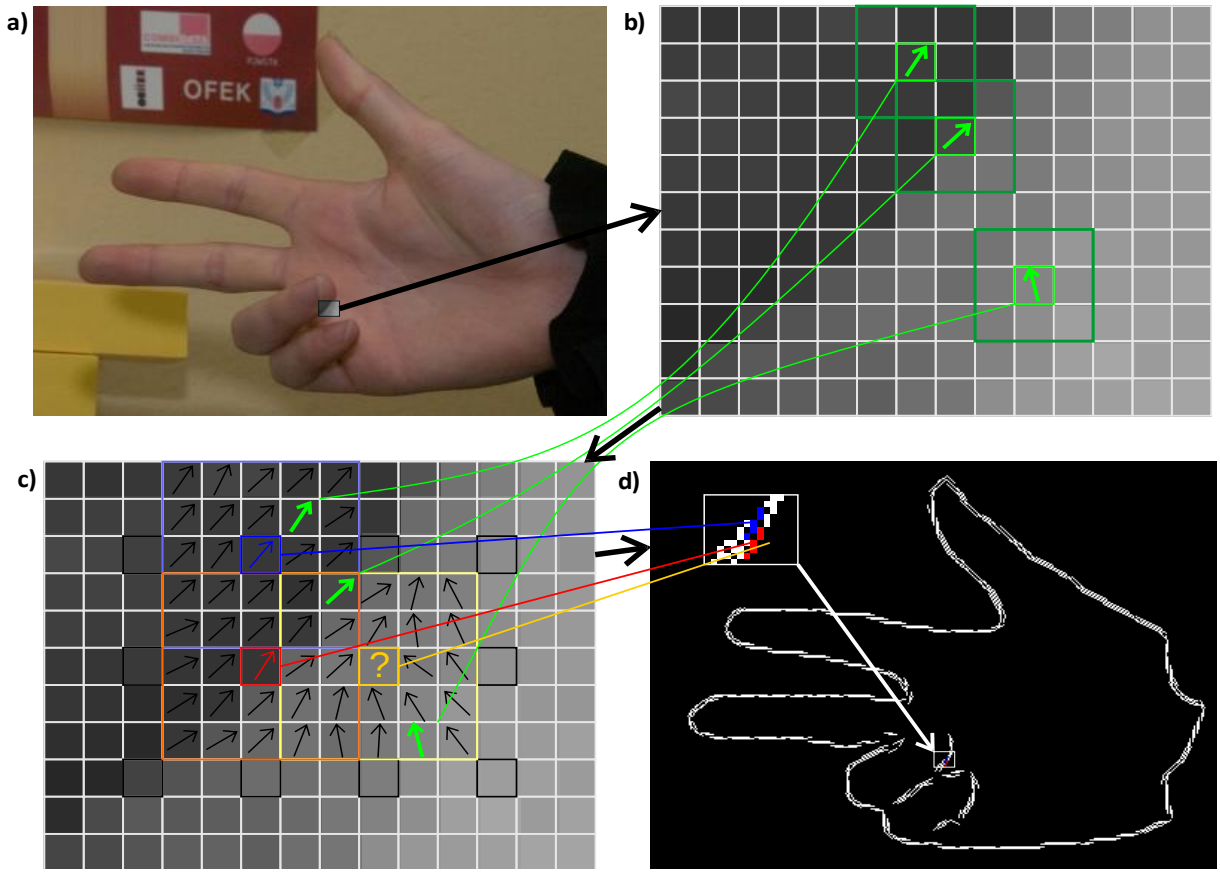
czyli dla macierzy 5×5. Wektor jednostkowy stycznej wyznaczany jest w taki sposób, aby minimalizował funkcję błędu wyznaczaną równaniem:

$$\delta(u_{m,n}) = \sum_{i=m-2}^{m+2} \sum_{j=n-2}^{n+2} (g_{i,j} \cdot u_{m,n})^2 . \tag{8}$$

Funkcja jest minimalizowana analitycznie, aby otrzymać kąt stycznej. Jeżeli wariancja kierunków gradientu jest wysoka wewnątrz grupy pikseli, znormalizowana funkcja błędu uśredniania zdefiniowana za pomocą równania:

$$E_{m,n} = \frac{\delta_{min}(u_{m,n})}{\sum_{i=m-2}^{m+2} \sum_{j=n-2}^{n+2} (g_{i,j})^2} , \tag{9}$$

uzyskuje wysoką wartość, a co za tym idzie, wyznaczony kąt nie jest wiarygodny. Dlatego na końcowym obrazie kierunkowym pojawiają się takie punkty, dla których błąd jest niski, czyli dla których wartość wagi dopasowania jest wysoka. Waga określona jest jako suma kwadratów gradientów w sąsiedztwie.



Rys. 16 Kolejne etapy uzyskiwania obrazu kierunkowego. b) Zbliżenie wskazanego na obrazie źródłowym a), regionu wraz z wizualizacją wyliczenia gradientu dla 3 wybranych pikseli. c) Wyznaczenie kąta stycznej dla co trzeciego piksela w co trzecim rzędzie, korzystając z 25 gradientów. Czerwona oraz niebieska strzałka zostaje zwizualizowana jako odcinek na obrazie kierunkowym d). W macierzy 5×5 oznaczonej żółtym kolorem, wariacja jest zbyt wysoka, co powoduje brak wizualizacji odcinka w tych współrzędnych.

Na tym etapie, dla co trzeciego piksela w co trzecim rzędzie znany jest gradient przedstawiony w postaci biegunowej. We wskazanych współrzędnych znane są kąt oraz waga gradientu. Aby uzyskać obraz kierunkowy, we współrzędnych, w których waga większa jest niż zdefiniowany próg, nanoszony jest odcinek o zadanej długości i wyliczonym kącie stycznej. Ponadto powyższe operacje przeprowadzane są tylko dla obszaru wewnątrz maski. Po odpowiednim ustaleniu progu, na obrazie kierunkowym widać kontur dłoni oraz kontury wewnętrzne ukazujące zgięte palce. Rys. 7 przedstawia pełną ilustrację pozyskiwania obrazu kierunkowego. Przykład pokazuje pozyskiwanie operatorów Sobela w otoczeniu 3×3 dla kilku wybranych pikseli zaznaczonych na zielono, a następnie dla wybranych co dziewiątych pikseli liczony jest kąt oraz waga gradientu. Czerwona i niebieska strzałka na Rys. 7 c posiada odpowiednią wagę, aby nanieść odcinek na obraz kierunkowy, natomiast przykład żółty ma zbyt duży błąd, dlatego nie jest brany pod uwagę w trakcie rysowania obrazu kierunkowego.

4. Propozycja autorskiego algorytmu wykrywania punktów charakterystycznych

Algorytm autorski został opracowany na podstawie zaproponowanego zbioru reguł decyzyjnych. Badania algorytmów wnioskujących opisano w publikacjach [90, 91, 92].

Analiza zaimplementowanych istniejących rozwiązań i metod wykrywania punktów charakterystycznych oraz wstępne określenie ich głównych wad i zalet pozwoliła na wyznaczenie kierunków badań mających na celu opracowanie autorskiego algorytmu. Główną wadą prezentowanych algorytmów jest wykorzystanie jedynie maski dłoni jako obrazu wejściowego metod wyznaczania punktów charakterystycznych. Sama maska dłoni jest jedynie regionem określającym miejsce występowania ręki na obrazie kolorowym. Nie niesie ona informacji o tym, co znajduje się wewnątrz niej. Dlatego postanowiono rozszerzyć dane wejściowe do algorytmu analizując nie tylko maskę dłoni, ale również obraz kierunkowy, który dokładniej opisano w rozdziale 3.2.4.

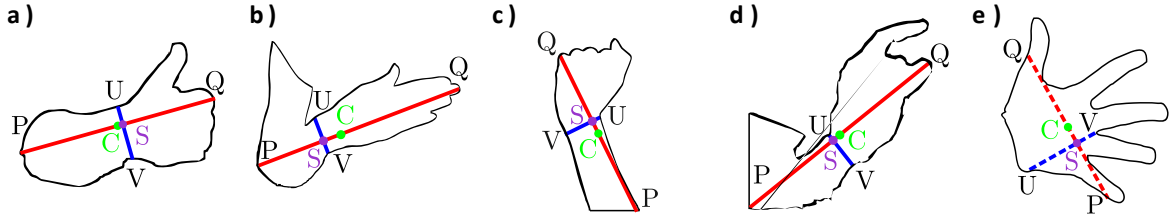
Algorytm składa się z trzech głównych kroków. Pierwszym etapem jest określenie orientacji dłoni oraz wyznaczenie regionu nadgarstka. Linia przechodząca przez dwa punkty nadgarstka dzielić będzie obraz dłoni na dwie części: region dłoni oraz pozostała część ręki. Następnie wykonywany jest szereg funkcji przetwarzania obrazu mający na celu wyłonienie zbioru punktów charakterystycznych. W ostatnim etapie punkty te zostają sklasyfikowane jako konkretne punkty charakterystyczne dłoni lub odrzucone jako fałszywe.

W poniższych podrozdziałach opisano dokładnie przebieg działania algorytmu. Przedstawiono metody sztucznej inteligencji i wnioskowania determinujące rodzaj i klasyfikujące wykryte punkty do poszczególnych grup.

4.1. Wykrywanie i lokalizacja punktów nadgarstka

Przedstawiona poniżej metoda wykrywania nadgarstka została przebadana oraz opisana w publikacjach [76, 77]. Opiera się ona na następujących zaobserwowanych regułach:

- Odcinek wyznaczający orientację ręki w przybliżeniu przebiega od łokcia do nadgarstka i jest to najdłuższa cięciwa konturu utworzonego z maski dłoni oraz zawierająca się wewnątrz niej.
- Linia zawierająca odcinek nadgarstka jest prostopadła do linii orientacji dłoni i występuje w regionie lokalnego zwężenia konturu utworzonego z maski dłoni.



Rys. 17 a-d) Przykłady konturów utworzonych z masek dłoni z zaznaczonymi odcinkami nadgarstka oraz orientacji dłoni. e) Przykład konturu utworzonego z maski nie spełniającego założeń.

Oba założenia nie sprawdzają się w przypadku zdjęć ręki z długim rękawem lub w przypadkach ułożenia ręki pod nietypowym kątem.

Za pomocą maski dłoni tworzony jest kontur regionu ręki. Wewnątrz konturu odnajdywana jest najdłuższa cięciwa *PQ* oraz jej punkt środkowy *S*. Prostopadle do odcinka *PQ* odnajdywany jest taki odcinek *UV*, którego długość jest najkrótsza, zakładając, że punkty *U* i *V* znajdują się na konturze. Odcinki *UV* oraz *PQ* przecinają się w punkcie *S*. Punkty *U* i *V* są to punkty nadgarstka, czyli pierwsze wykryte punkty charakterystyczne dłoni.

W celu odnalezienia punktów nadgarstka opracowano algorytm działający zgodnie z powyższym opisem oraz zaobserwowanymi faktami. W formie skróconej algorytm można przedstawić w następujący sposób:

Algorytm 4: Poszukiwanie punktów nadgarstka

1. Dla podanej maski dłoni wyznacz kontur.
2. Wyznacz najdłuższą przekątną konturu.
3. Obróć obraz o kąt nachylenia prostej przechodzącej przez punkty PQ.
4. Wyznacz profil obrazu, lokalne ekstrema oraz punkt nadgarstka w minimum lokalnym.
5. Zorientuj wykryte punkty w obrazie wejściowym.

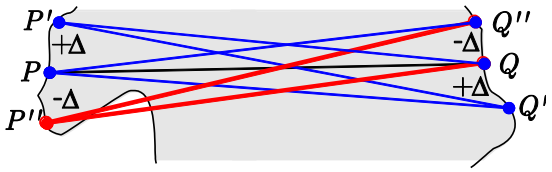
Wizualizacją maski dłoni jest obraz binarny. Łatwo zatem wyznaczyć kontur dłoni, na przykład stosując operacje morfologiczne. Reprezentacją konturu jest usystematyzowany ciąg punktów brzegowych regionu dłoni.

Najdłuższa przekątna potrzebna jest do wyznaczenia kąta obrotu obrazu. Dlatego wyznaczenie współrzędnych odcinka *PQ* potrzebne jest jedynie w celu ustalenia kąta nachylenia prostej przechodzącej przez te punkty względem osi x. W celu odnalezienia najdłuższej przekątnej należałoby przeszukać wszystkie odcinki pomiędzy każdą parą punktów konturu, wyeliminować takie, które nie zawierają się wewnątrz konturu oraz wyłonić najdłuższy z nich. Metoda taka nazywana jest algorytmem siłowym (ang. brute force). Niestety, metoda ta posiada wysoką złożoność obliczeniową, ponieważ wymaga analizy każdej pary punktów ze zbioru punktów konturu. Jeżeli liczebność zbioru punktów konturu przedstawi się jako *A*, wtedy liczba kombinacji dwuelementowych wyniesie:

$$C_n^k = \binom{n}{k} = \frac{n!}{k!\,(n-k)!} = \frac{A!}{2!\,(A-2)!} = \frac{1}{2}A^2 - A\,, \qquad (10)$$

gdzie C jest kombinacją bez powtórzeń n elementów z k elementowego zbioru. Wykazano zatem, że złożoność obliczeniowa rośnie wielomianowo wraz ze wzrostem liczby punktów konturu. Propozycją zmniejszenia złożoności tej części algorytmu jest analiza jedynie części punktów i wyłonienie odcinka, który jest zbliżony do najdłuższej cięciwy. Pary punktów dobierane są losowo w każdej iteracji, długość odcinka porównywana jest z najdłuższą do tej pory odnalezioną długością oraz sprawdzany jest warunek, czy odcinek zawiera się wewnątrz konturu. Jeżeli odnaleziona została para punktów, która jest dłuższa od wcześniej wykrytych, para ta jest zapamiętywana. Jeżeli po upływie zadeklarowanej ilości iteracji *L,* po

odnalezieniu najdłuższej przekątnej, nie została odnaleziona żadna dłuższa uznaje się, że wybrane punkty tworzą najdłuższą przekątną. Po odnalezieniu tych punktów przeprowadzane jest poprawianie lokalne. Punkty *P* i *Q* przesuwane są wzdłuż konturu o mały krok *Δ*. Jeżeli odcinek *PQ* jest krótszy niż jeden z nowo powstałych odcinków, nowe punkty zapisywane są jako *PQ*, a poprawianie lokalne przeprowadzone jest ponownie.



Rys. 18 Poprawianie lokalne etapu poszukiwania najdłuższej przekątnej. Kolorem niebieskim oznaczone są potencjalne nowe przekątne, a czerwonym odcinki nie spełniające warunku zawierania się wewnątrz maski.

Co więcej, mimo małej popularności badań odnośnie wykrywania punktów nadgarstka, odnaleziono inne rozwiązanie odnajdywania orientacji dłoni, z wykorzystaniem momentów obrazu [75]. Momenty obrazu maski są to specyficzne wartości wagowe natężenia grupy pikseli. Obraz można przedstawić jako funkcję dwóch zmiennych *f(x,y)*. Moment *M* rzędu *(p+q)* zdefiniowany jest jako

$$M_{pq} = \sum_x \sum_y x^p y^q f(x,y)\,. \tag{11}$$

Wyznaczanie kolejnych momentów obrazu pomaga przy określeniu wielu przydatnych miar. Przykładowo pole powierzchni obiektu przedstawionego na obrazie binarnym jest równe momentowi M_{00}. Moment centralny μ jest to rozkład prawdopodobieństwa zmiennej losowej wokół wartości oczekiwanej. Wyznaczając kolejne wartości momentu centralnego opisanego wzorem:

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y)\,, \tag{12}$$

Otrzymujemy między innymi:

$$\begin{aligned}
&\mu_{00} = M_{00}\,,\\
&\mu_{01} = \mu_{10} = 0\,,\\
&\mu_{11} = M_{11} - \bar{x}M_{01} = M_{11} - \bar{y}M_{10}\,,\\
&\mu_{20} = M_{20} - \bar{x}M_{10}\,,\\
&\mu_{02} = M_{02} - \bar{y}M_{01}\,.
\end{aligned} \tag{13}$$

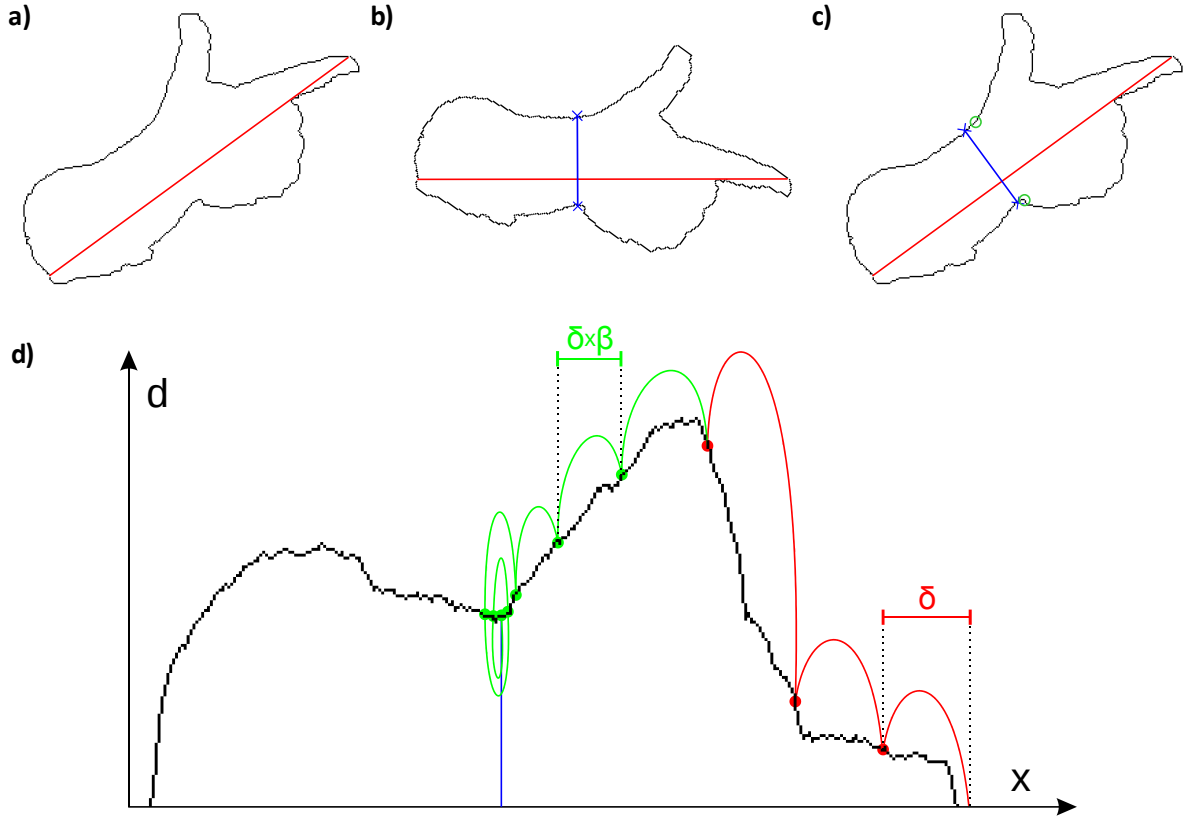
Momenty centralne wykorzystywane są między innymi do wyznaczenia orientacji danego obszaru. Orientacja obiektu zapisana jako kąt, liczona jest wzorem:

$$\alpha = \frac{1}{2}\tan^{-1}\left(\frac{2\mu'_{11}}{\mu'_{20} - \mu'_{02}}\right). \qquad (14)$$

W ten sposób w [75] opisano metodę wyznaczania kąta nachylenia linii wyznaczającej orientację dłoni.

Kolejnym etapem jest znalezienie odcinka prostopadłego do linii orientacji dłoni. W tym celu wyznaczany jest kąt orientacji dłoni poprzez odnalezienie najdłuższej przekątną albo za pomocą momentów. Następnie obraz obracany jest względem punktu *C* w taki sposób, aby dłoń zorientowana była poziomo. Dzięki temu można wyliczyć profil dłoni, czyli dla danej kolumny macierzy obrazu, wyliczana jest odległość pomiędzy występującymi punktami konturu. Zatem wartość funkcji profilu dla danej kolumny odwzorowuje długość cięciwy poprowadzonej w tej kolumnie. Analizując funkcję profilu w poszukiwaniu globalnego maksimum, a następnie lokalnego minimum, można wyznaczyć dokładne miejsce przewężenia konturu, czyli współrzędne punktów *UV*.

Zaobserwowano, że profil dłoni posiada zazwyczaj dwa główne maksima. Jedno w regionie dłoni, drugie w okolicach przedramienia. Nadgarstek znajduje się natomiast w okolicy lokalnego minimum pomiędzy tymi maksimami. Dlatego nie ma znaczenia, czy lokalne minimum odnajdywane będzie po prawej, czy po lewej od pierwszego napotkanego lokalnego maksimum. W prezentowanych badaniach przyjęto metodę poszukiwania ekstremów od prawej strony do lewej.



Rys. 19 Kolejne etapy wykrywania punktów nadgarstka. a) Odnalezienie najdłuższej cięciwy konturu, d) wyznaczenie funkcji profilu obróconego obrazu b), d) naniesienie wykrytych punktów na obraz oryginalny.

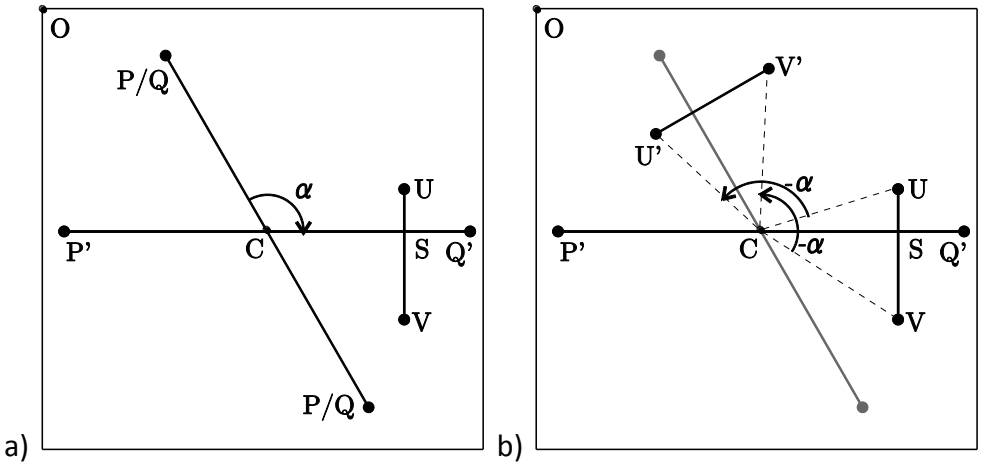
Rys. 19 przedstawia kolejne etapy poszukiwania punktów nadgarstka. Po obróceniu obrazu o wyznaczony kąt (Rys. 19 b), czyli tak, aby najdłuższa cięciwa zorientowana była poziomo, wyznaczany jest profil przedstawiony na Rys. 19 d. Profil jest funkcją odległości *d* dwóch punktów konturu dla każdej współrzędnej *x* obrazu obróconego. Funkcja odległości posiada maksimum globalne w regionie wzniesionego kciuka. W sąsiedztwie kolumn zawierających piksele kciuka odnaleziono pierwsze lokalne maksimum. Następnie, poszukiwane jest pierwsze lokalne minimum. W miejscu, w którym zostało ono odnalezione, zaznaczono odcinek pionowy na Rys. 19 d. Skrajne punkty zaznaczone krzyżykami oznaczają punkty nadgarstka i są one przeniesione na Rys. 19 c, na którym widać zaznaczone krzyżykami wykryte punkty nadgarstka oraz punkty wzorcowe, pobrane z bazy, reprezentowane przez koła.

Wyznaczenie kolejnych wartości odległości punktów konturu dla każdej kolumny obrazu wymaga wielokrotnego odwołania się do elementów macierzy obrazu. W celu optymalizacji czasu zaproponowano algorytm poszukiwania maksimum i minimum wzdłuż kierunku z kontrakcją geometryczną, modyfikując algorytm [93]. Krok poszukiwań δ ustalony jest na 10% szerokości obrazu. Przemieszczając się w kierunku od prawej do lewej poszukiwana jest pierwsza wartość, która jest mniejsza od poprzedniej. Oznacza to, że poprzednia wartość występowała w regionie maksimum. Ten krok widnieje na Rys. 19 d w formie skoków oznaczonych czerwoną krzywą. Następnie, zgodnie z algorytmem poszukiwania minimum z kontrakcją geometryczną, w przypadku jeżeli następny punkt ma większą wartość niż poprzedni, zmieniany jest kierunek poszukiwań i zmniejszany krok poprzez przemnożenie go przez wartość β. Ścieżka poszukiwania minimum zwęża się kilka razy i odbija, tak aby ostatecznie, w przypadku gdy krok jest mniejszy niż żądana dokładność, określić punkt, w którym profil ma lokalnie najmniejszą wartość. Poszukiwanie minimum oznaczone jest krzywą zieloną na Rys. 19 d. Zmiana kierunku poszukiwań reprezentowana jest przejściem krzywej przez wykres. Poszukiwanie wzdłuż kierunku w prawo reprezentuje krzywa pod wykresem, a w lewo nad wykresem. Dodatkowo warto zauważyć, że w prezentowanym przykładzie minimum zostało odnalezione po zaledwie 10 krokach, czyli zaledwie 10 razy wywoływana była operacja liczenia odległości punktów konturu w danej kolumnie.

Po wykryciu punktów w obrazie obróconym, punkty należy nanieść na obraz pierwotny. W tym celu stosowany jest wzór na obrót punktu o kąt. Wzór ten jednak należy odnieść do nietypowego układu współrzędnych w obrazie, gdzie punkt *O* jest punktem *(0,0)*, a wartości *x* i *y* rosną w prawo i w dół. Aby wyznaczyć punkty *U'* i *V'* należy zastosować równanie:

$$\begin{aligned} x_{U'} &= x_S \cos(\propto) - y_U \sin(\propto) + x_C \\ y_{U'} &= -x_S \sin(\propto) - y_U \cos(\propto) + y_C \end{aligned} \tag{15}$$

Punkt *V'* liczony jest tak samo jak punkt *U'*. Aby wyznaczyć te punkty potrzebny jest indeks kolumny, w której odnaleziono minimum, czyli wartość x_S, oraz należy pamiętać, aby zmodyfikować wzór na obrót punktu o kąt względem zmodyfikowanego układu współrzędnych i względem punktu *C*.

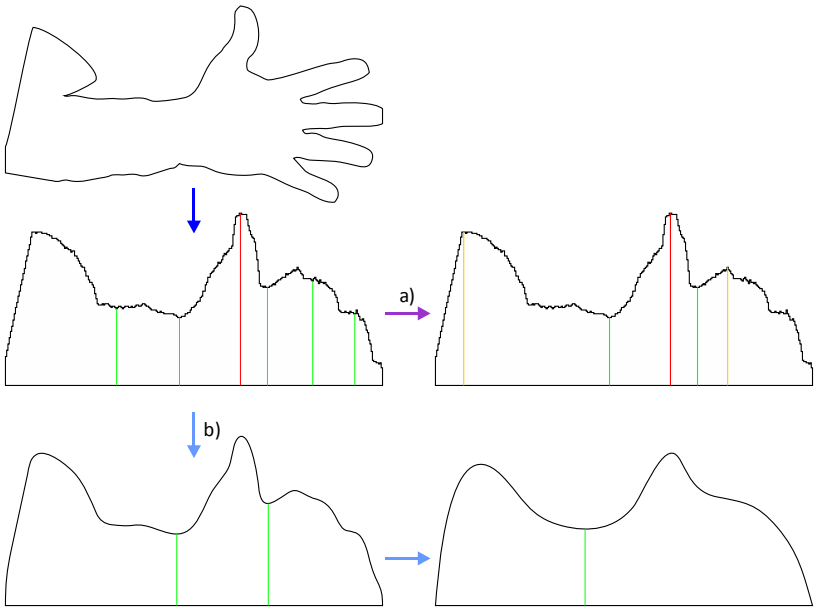


Rys. 20 Proces orientacji wykrytych punktów: a) obracanie obrazu zgodnie z kątem nachylenia najdłuższej cięciwy PQ, b) obracanie punktów nadgarstka UV nanosząc je w odpowiednie miejsca w obrazie pierwotnym

Wyznaczone punkty *U'* i *V'* są jeszcze dodatkowo identyfikowane z dokładnymi punktami w wektorze punktów konturu oraz z ich indeksami. Algorytm poszukiwania nadgarstka kończy się odnalezieniem punktów *U'* i *V'*, czyli punktów nadgarstka.

4.1.1. Propozycje ulepszenia metody wykrywania i lokalizacji punktów nadgarstka

W trakcie testów i implementacji algorytmu wykrywania punktów charakterystycznych zebrano zbiór wniosków odnośnie sposobu działania algorytmu. W celu poprawy algorytmu opisanego w rozdziale 4.1 zaproponowano dodatkowe metody odnajdywania optymalnego miejsca minimum lokalnego, które uznawane jest za punkt nadgarstka.



Rys. 21 Sposób pozyskania profilu maski dłoni z zaznaczonym globalnym maksimum (czerwony) oraz kilkoma lokalnymi minimami (zielony). a) Metoda polegająca na odnalezieniu najlepszego z minimów lokalnych, b) metoda filtracji profilu

Po pierwsze zauważono, że do wykrytych punktów nadgarstka można zastosować poprawianie lokalne, działające w podobny sposób co poprawianie lokalne najdłuższej

przekątnej opisywane przy okazji Rys. 18. Sama ta operacja nie zmienia znacząco sposobu działania algorytmu, jednak w znaczący sposób obniża błąd dopasowania punktów wykrytych poprawnie.

Zakładając, że metoda gradientowa połączona z kontrakcją przedziału może „przeskoczyć" minimum lub może być zależna od strony, z której zaczęto analizę, zaproponowano metodę niezależną od kierunku poszukiwań. Analizując Rys. 21, po uzyskaniu profilu z maski dłoni, idąc ścieżką *a,* poszukujemy globalnego maksimum profilu oznaczonego kreską czerwoną. Następnie po obu stronach tego maksimum odnajdujemy kolejne dwa maksima globalne, zaznaczone kreską pomarańczową. Zakłada się, że jedno z tych maksimów będzie w regionie zgrubienia śródręcza, a drugie lokalnego zgrubienia w wystających palcach. Pomiędzy dwoma parami trzech wykrytych maksimów szukane są dwa minima lokalne oznaczone na zielono. Ostatecznie wybierane jest to minimum, którego wartość najbardziej zbliżona jest do statystycznie określonej miary.

Kolejne podejście wynika z obserwacji struktury profilu. Zauważono, że jego wykres jest bardzo poszarpany. Oznacza to, że profil posiada wiele minimów lokalnych. Aby określić, które z minimów powinno być wybrane, funkcja profilu poddawana jest filtracji medianowej o jednym, iteracyjnie rosnącym wymiarze. Profil filtrowany jest tak długo, aż posiada tylko jedno minimum lokalne, oczywiście nie włączając minimów brzegowych równych 0.

Niestety zgodnie z wynikami uzyskanymi w sekcji 5.3.1 okazało się, że zaproponowane metody nie poprawiają jakości wykrycia, co dodatkowo pokazuje, iż metoda opisana w rozdziale 4.1 z poprawianiem lokalnym jest najskuteczniejszą znaną metodą odnajdywania punktów nadgarstka.

4.2. Wykorzystanie wybranych metod analizy obrazu do lokalizacji punktów charakterystycznych

Po przeanalizowaniu wszystkich wad i zalet zaproponowanych w literaturze rozwiązań poszukiwania nadgarstka, zauważono sporą wadę wszystkich rozwiązań bazujących na masce dłoni. Mianowicie maska dłoni niesie informacje jedynie o regionie występowania skóry i bezpośrednio nie przedstawia struktury dłoni wewnątrz niej. Dlatego głównym celem opisanej metody jest uzyskanie informacji o tym, co znajduje się wewnątrz maski. Dodatkowo zaobserwowano następujące reguły:

- Obraz kierunkowy wyznaczony wewnątrz regionu skóry przenosi informacje o ułożeniu i pozycji palców, które są zgięte lub znajdują się wewnątrz maski;
- Obraz kierunkowy przedstawia cząstkowe kontury palców, które wykazują podatność na lokalizację ich za pomocą algorytmu zbliżonego do dopasowania wzorca z rozdziału 2.3.3;
- Algorytm wykorzystujący transformacie odległościową, opisany w rozdziale 2.3.2, posiada potencjał do identyfikacji wykrytych regionów dłoni;

W celu przeprowadzenia algorytmu poszukiwania punktów charakterystycznych należy przygotować szereg obrazów, które są wynikami metod przetwarzania obrazu opisanych w

rozdziale 3.2. W celu wyjaśnienia sposobu działania algorytmu, zaprezentowane zostaną przykłady przetwarzania obrazu z obrazu wejściowego prezentowanego na Rys. 22.

a) b)

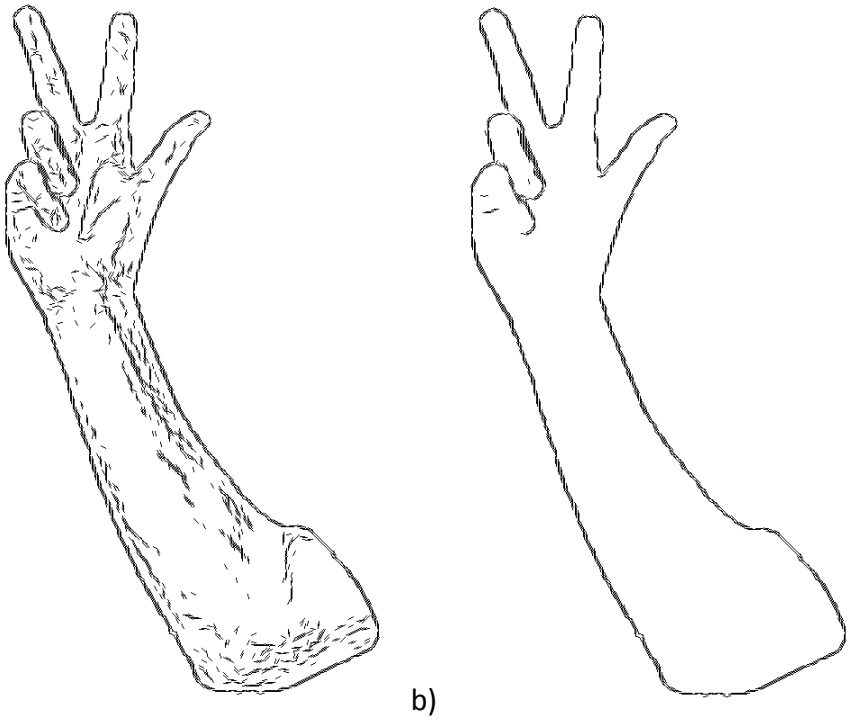
Rys. 22 Przykład z bazy gestów. a) Obraz barwny przedstawiający gest „3", b) maska z regionem skóry.

Celem tego etapu jest wyznaczenie obrazu w postaci mapy prawdopodobieństwa występowania punktów charakterystycznych. Algorytm przedstawiony jest jako:

Algorytm 5: Poszukiwanie punktów charakterystycznych

1. Dla podanego przez maskę regionu w obrazie kolorowym wyznacz obraz kierunkowy z określonym progiem.
2. Przeprowadź poszukiwanie wzorca koła na obrazie maski oraz wzorca okręgu na obrazie kierunkowym.
3. Połącz wyniki dopasowania wzorca.
4. Znajdź lokalne maksima w pozyskanym obrazie.

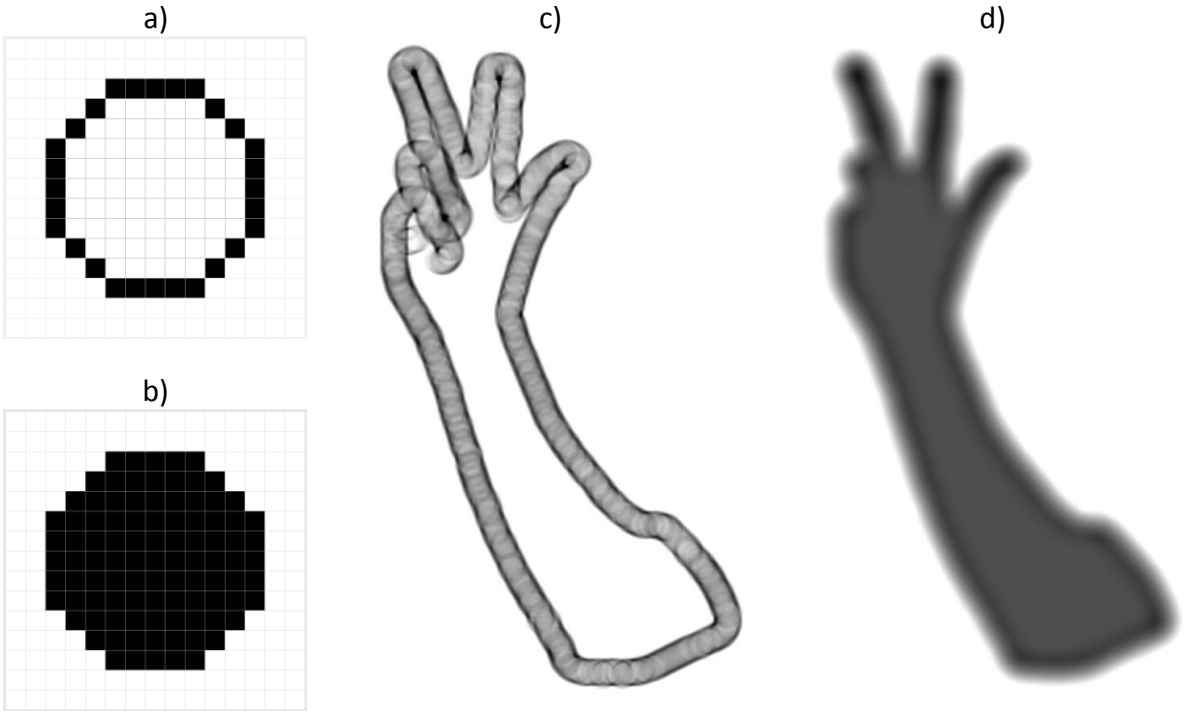
Pierwszym etapem jest wyznaczenie obrazu kierunkowego progowanego. Oznacza to, że dla obrazu kolorowego, po wyznaczeniu regionu skóry odcinane jest tło. Następnie wyznaczony jest obraz kierunkowy dla regionu występowania skóry. Dla każdego piksela wyznaczany jest kierunek gradientu lokalnego oraz jego intensywność. Wartości te reprezentowane są na obrazie przez odcinki o zadanej długości, występujące lub niewystępujące w zależności od zadanego progu, który filtruje wektory o wysokiej wariancji kierunków gradientu, czyli wysokiej wartości błędu wyliczanej z użyciem wzoru (9). Próg powinien być tak dobrany, aby obraz docelowy miał jak najlepiej widoczne kontury palców, bez zbędnych informacji o strukturze skóry, zagięć lub paznokci.



a) b)

Rys. 23 a) Obraz kierunkowy wewnątrz maski dla obrazu kolorowego, b) Obraz kierunkowy progowany.

Na tym etapie widać już przewagę algorytmu wykorzystującego obraz kierunkowy (Rys. 23 b) względem algorytmów bazujących jedynie na masce dłoni (Rys. 22 b). Na obrazie maski dłoni nie widać bezpośrednio ułożenia dwóch zagiętych palców.



Rys. 24 c) Wynik dopasowania a) wzorca okręgu do obrazu kierunkowego (Rys. 23 b).
d) Wynik dopasowania b) wzorca koła do maski dłoni (Rys. 22 b).

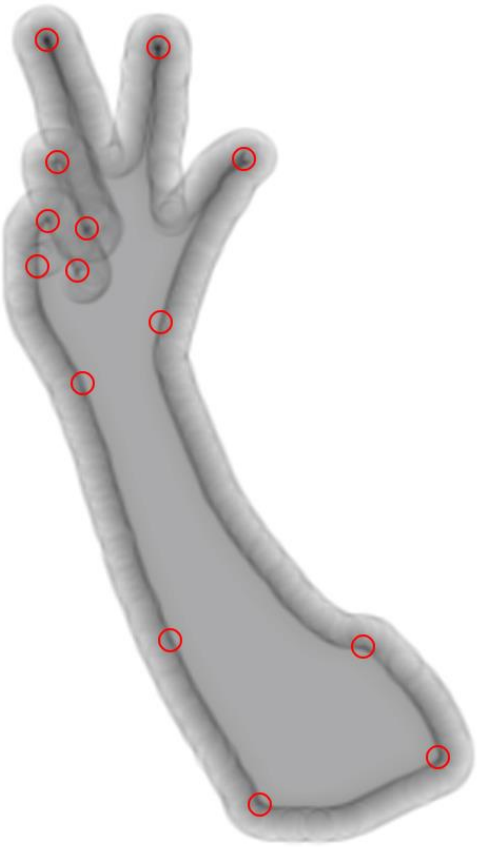
Kolejnym krokiem jest wyznaczenie obrazów mapy prawdopodobieństwa występowania wzorca koła oraz okręgu. Zgodnie z metodą poszukiwania wzorca z użyciem wzajemnej korelacji, która opisana została w rozdziale 3.2.3, przeprowadzane są dwie operacje

dopasowania wzorca. Pierwsza, jako obraz źródłowy traktuje obraz kierunkowy, do którego dopasowywany jest obraz okręgu. Ważne jest, aby średnica koła odpowiadała średnicy palca. W tym celu rozmiar wzorca jest ustalany proporcjonalnie do wielkości dłoni. Jak można zaobserwować na Rys. 24, koncentracja wartości maksymalnych znajduje się na czubkach palców, nawet tych zagiętych, oraz wzdłuż palców.

Jednak maksima występują również pomiędzy palcami. Aby wykluczyć te regiony, wyznaczana jest również mapa prawdopodobieństwa dopasowania wzorca koła do maski dłoni. Dla takiego obrazu koncentracja wartości maksymalnych znajduje się jedynie na czubkach palców, natomiast w przestrzeni między palcami znajduje się wartość bliska 0. Połączenie tych dwóch obrazów sprawia, że wyostrzone są punkty czubków palców, zachowane są punkty zgiętych palców wewnątrz dłoni, oraz odrzucane punkty pomiędzy palcami.

Prezentowane obrazy zostały poddane negacji i poprawie kontrastu w celu uzyskania przejrzystszej prezentacji. W prezentowanych obrazach ciemniejszy kolor reprezentuje piksel o większej wartości prawdopodobieństwa.

Po zsumowaniu i znormalizowaniu obu macierzy prawdopodobieństwa otrzymano połączony obraz mapy dopasowania wzorów, przedstawiony na Rys. 25. Przeprowadzając test poszukiwania lokalnego można wykryć zbiór punktów charakterystycznych. Niestety, większość wykrytych punktów będzie nadal punktami fałszywymi, które należy odrzucić. Ponadto pozostałe, nie odrzucone punkty, należy poprawnie sklasyfikować.

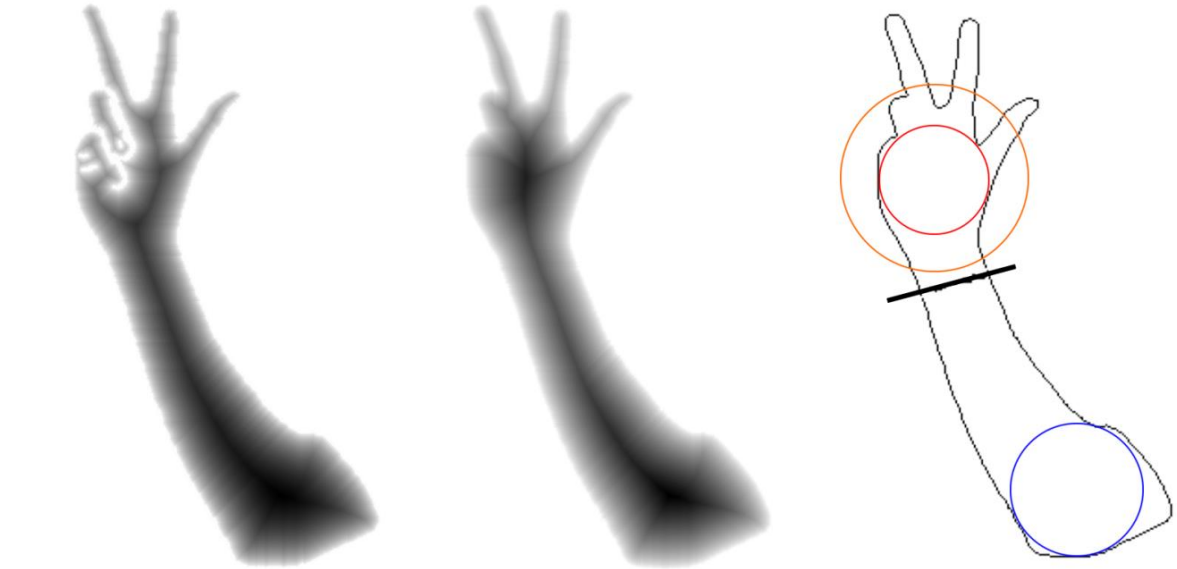


Rys. 25 Połączenie obu map prawdopodobieństwa dopasowania wzorca. Czerwonymi punktami zaznaczono przykładowe miejsca wykrycia maksimów lokalnych.

Przed przystąpieniem do klasyfikacji wykrytych punktów charakterystycznych należy przygotować jeszcze dwa obrazy transformaty odległościowej zgodnie z metodą opisaną w rozdziale 3.2.2.

Pierwszym obrazem jest transformata odległościowa obrazu kierunkowego. Obraz ten wykorzystywany jest w późniejszym etapie do ustalenia dokładnego regionu palca, ponieważ jak łatwo zaobserwować na Rys. 26 a, lokalne maksima transformaty odległościowej ulokowane są w rejonie palców.

Drugim obrazem jest transformata odległościowa maski dłoni. Wartość i lokalizacja maksimum transformaty odległościowej bezpośrednio przekłada się na promień i środek okręgu wpisanego w kontur. Dzięki tej właściwości uzyskać można region śródręcza, który w dalszych etapach potrzebny jest do klasyfikowania wykrytych punktów charakterystycznych. Dodatkowo region śródręcza ustalany jest na początku, ponieważ w proporcjonalny sposób przekłada się na średnice wykrywanego wzorca koła i okręgu. Operacja poszukiwania maksimum przeprowadzana jest po obu stronach linii rozdzielającej maskę na dwie części w miejscu nadgarstka, ponieważ na obecnym etapie nie wiadomo, która część to region dłoni, a która przedramienia. Na Rys. 26 c zaznaczono czerwonym i niebieskim okręgiem potencjalne miejsce regionu śródręcza, oraz na pomarańczowo region śródręcza przeskalowany o współczynnik proporcjonalny, opisany w rozdziale 5.2.3

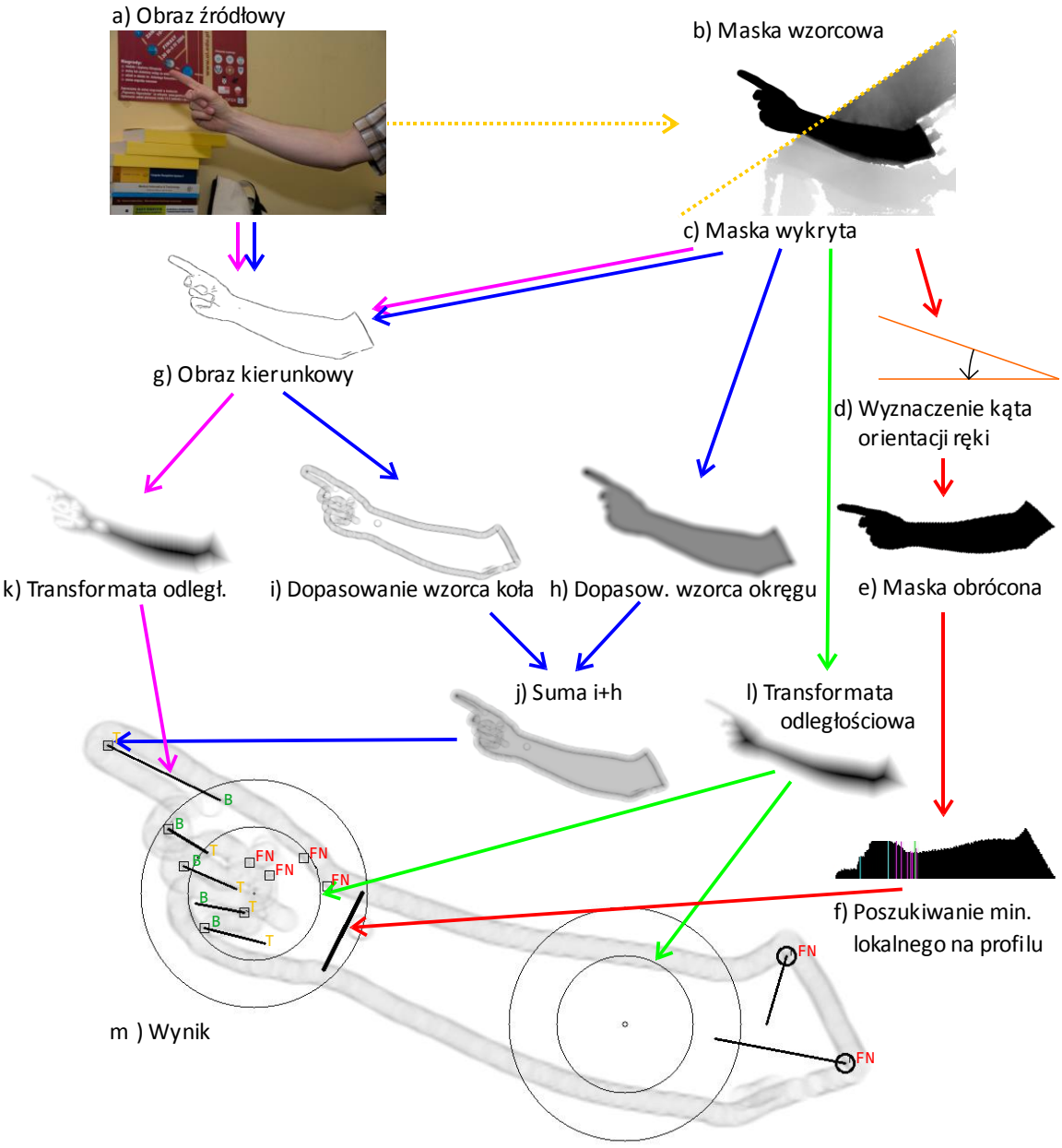


a) b) c)

Rys. 26 Wynik transformacji odległościowej a) dla obrazu kierunkowego, b) dla maski dłoni. c) Analiza transformaty odległościowej maski dłoni w celu odnalezienia regionu śródręcza.

Pełna procedura algorytmu autorskiego przedstawiona jest na Rys. 27. Dane wejściowe przedstawione są jako: obraz barwny (a) oraz wzorcowa (b) lub wykryta (c) maska dłoni. Procedura wykrycia nadgarstka oznaczona czerwoną ścieżką składa się z wyznaczenia kąta obrotu (d), obrotu maski dłoni (e) oraz odnalezienia lokalnego ekstremum na profilu dłoni (f). Ścieżka niebieska przedstawia procedurę odnajdywania punktów charakterystycznych, zawierającą obraz kierunkowy (g), wynik poszukiwania wzorca koła (h), wynik poszukiwania wzorca okręgu (i) oraz znormalizowaną sumę obrazów i+h (j). Wyznaczenie palców,

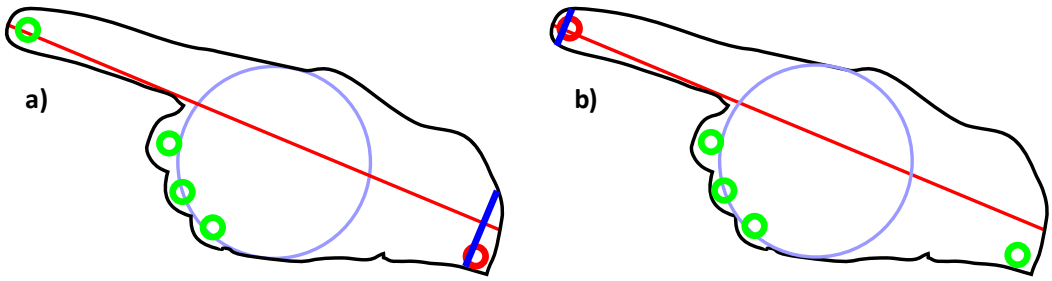
przedstawiono ścieżką fioletową, bazując na obrazie transformaty odległościowej (k) z obrazu kierunkowego. Region śródręcza wyznaczany jest na obrazie transformaty odległościowej (l) z maski dłoni. Ostateczna wizualizacja przedstawiona jest na Rys. 27 m. Wynikiem algorytmu autorskiego jest zbiór współrzędnych punktów charakterystycznych, przedstawionych w formie pliku XML.



Rys. 27 Pełna procedura algorytmu autorskiego.

Należy zwrócić jednak uwagę na różnicę między obrazami dłoni z długim oraz z krótkim rękawem. Krótki rękaw powoduje uwidocznienie regionu przedramienia, co zapewnia poprawne wykrycie nadgarstka, jednak wymusza analizę obu regionów w celu identyfikacji śródręcza i przedramienia. W takim przypadku poprawna lokalizacja nadgarstka jest niezbędna.

W przypadku obrazów reprezentujących samą dłoń, czyli rękę z krótkim rękawem, nadgarstek zostaje wykryty na brzegu konturu i nie dzieli on ręki na dwa regiony. Oznacza to, że koło wyznaczające region śródręcza ulokowane będzie w poprawnym miejscu niezależnie od wykrytego nadgarstka. Jednak błędnie określona pozycja nadgarstka może w negatywny sposób wpłynąć na klasyfikację niektórych wykrytych punktów.



Rys. 28 Przykłady konsekwencji wykrycia nadgarstka a) poprawnie oraz b) błędnie. Czerwone punkty odrzucane są, ponieważ znajdują się zbyt blisko niebieskiego odcinka nadgarstka.

Rys. 28 przedstawia kontur dłoni bez rękawa, na którym wykryto nadgarstek w miejscu faktycznego nadgarstka oraz w miejscu palca. Region śródręcza, oraz punkty charakterystyczne zostały wykryte w tych samych miejscach niezależnie od punktu nadgarstka. Jednak błędne wykrycie pozycji nadgarstka może skutkować odrzuceniem niewłaściwych punktów charakterystycznych.

4.3. Selekcja i klasyfikacja wykrytych punktów charakterystycznych

Wynikiem przeprowadzenia operacji poszukiwania maksimów lokalnych na połączonej macierzy map prawdopodobieństwa, jest zbiór potencjalnych punktów charakterystycznych, tak jak przedstawia przykład z Rys. 25. Selekcja przeprowadzana jest w celu odrzucenia punktów wykrytych w niewłaściwych współrzędnych. Pozostałe punkty poddawane są klasyfikacji w celu określenia ich przynależności do danej grupy punktów.

W celu przystąpienia do selekcji i klasyfikacji wykrytych punktów charakterystycznych należy przygotować następujące dane:

- Pozycja punktów nadgarstka;
- Pozycja dwóch potencjalnych regionów śródręcza wyznaczona z transformaty odległościowej po obu stronach nadgarstka;
- Połączenie obu map prawdopodobieństwa dopasowania wzorca;
- Obraz będący wynikiem transformaty odległościowej z obrazu kierunkowego.

Potencjalnymi punktami, które będą poddawane klasyfikacji są lokalne maksima macierzy prawdopodobieństwa dopasowania wzorców. Każdy wykryty punkt P o współrzędnych (x_P,y_P) poddawany jest procesowi klasyfikacji, który powstał z połączenia reguł wnioskowania oraz drzew decyzyjnych. Każdy punkt poddawany jest następującym podziałom:

- Wykryty punkt jest odrzucany jeżeli znajduje się zbyt blisko nadgarstka lub konturu. Jeżeli w obrazie transformaty odległościowej z maski dłoni (DT), punkt o tych samych współrzędnych ma wartość mniejszą niż zadany próg, należy odrzucić ten punkt. Region filtracji takich punktów oznaczony jest strefą koloru czerwonego na Rys. 29 oraz opisany wzorem:

$$DT_{P(x_P,y_P)} > t_{DT} \,. \tag{16}$$

- Jeżeli punkt został wykryty powyżej prostej wyznaczonej przez punkty nadgarstka, należy sklasyfikować go do grupy pierwszej, jeżeli poniżej, do grupy drugiej. Na tym etapie numer grupy nie oznacza przynależności punktów do region śródręcza lub regionu przedramienia. Region grupy pierwszej i drugiej oznaczony jest strefą koloru zielonego oraz niebieskiego, (zarówno ciemny jak i jasny) na Rys. 29. Przynależność do grupy określa się spełnieniem lub niespełnieniem nierówności:

$$(y_P - y_{W1})(x_{W2} - x_{W1}) - (y_{W2} - y_{W1})(x_P - x_{W1}) > 0 \,. \tag{17}$$

  P oznacza wykryty punkt, a *W1* i *W2* są to punkty nadgarstka. Wzór jest zmodyfikowaną wersją wzoru na prostą przechodzącą przez punkty *A* i *B*:

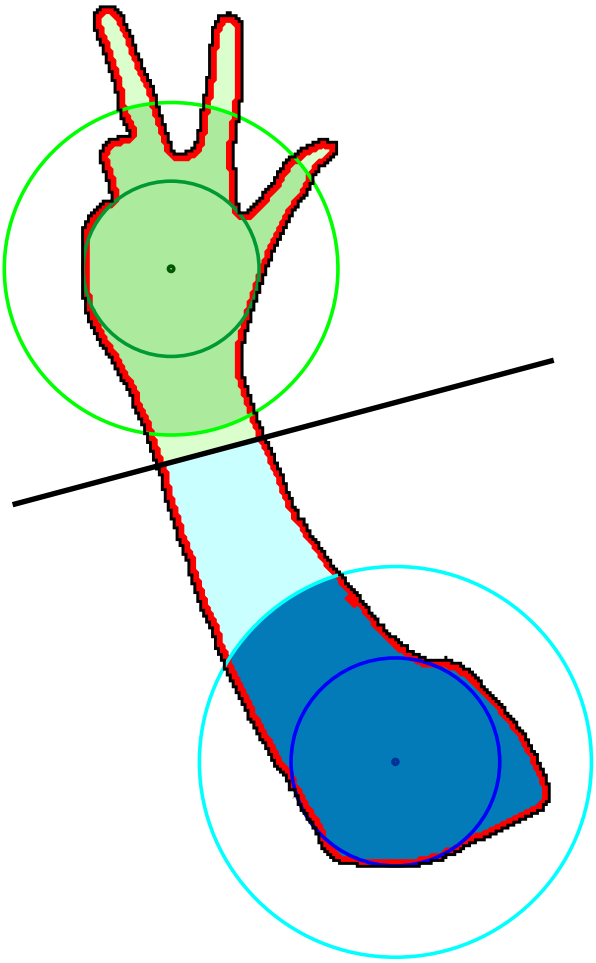
$$(y - y_A)(x_B - x_A) - (y_B - y_A)(x - x_A) = 0 \,. \tag{18}$$

- Jeżeli punkt wykryty został wewnątrz okręgu o promieniu pomnożonym przez współczynnik powiększenia strefy śródręcza λ. oznaczony jest on jako część palca zgiętego. Jeżeli wykryty jest na zewnątrz, oznaczony jest jako czubek palca wyprostowanego. Palce zgięte wykrywane będą wewnątrz ciemnego regionu, a palce wyprostowane wewnątrz jasnego na Rys. 29. Aby określić region w programie używa się funkcji wyznaczenia odległości wykrytego punktu *P* od środka regionu śródręcza *S*, a następnie porównuje się go do promienia przemnożonego o współczynnik powiększenia strefy śródręcza:

$$|PS| > r_S \cdot \lambda. \tag{19}$$

Wykryty punkt, jeżeli nie zostanie odrzucony, zostaje sklasyfikowany do grupy pierwszej lub grupy drugiej, w zależności od występowania poniżej lub powyżej punktu nadgarstka. W trakcie wykrywania kolejnych punktów, każda grupa zlicza liczbę punktów do niej należącej.

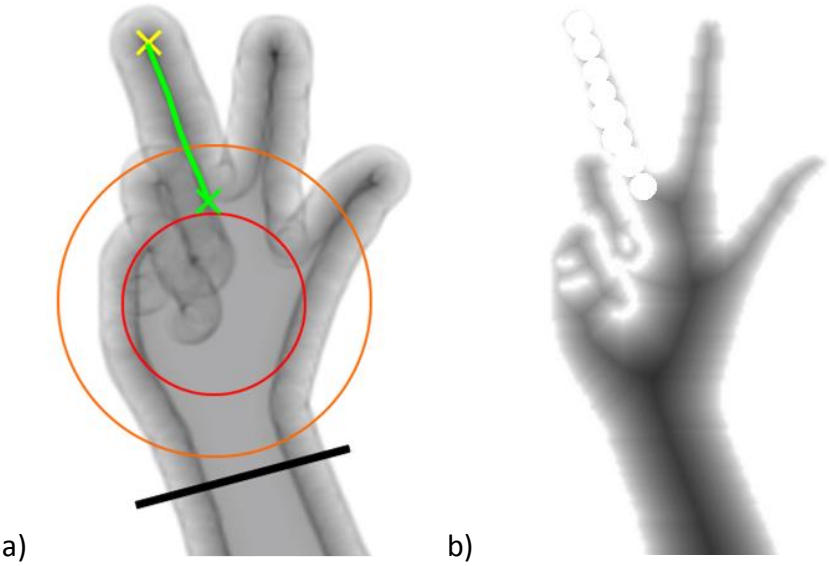
Skupiając się na ostatnim kryterium, przeanalizujmy co dzieje się w przypadku wykrycia punktu należącego do zgiętego lub wyprostowanego palca. Nowo wykryty punkt *P*, jeżeli znajduje się poza okręgiem o promieniu pomnożonym przez współczynnik powiększenia strefy śródręcza λ, traktowany jest jako wyprostowany. Palec wyprostowany ma nasadę na granicy strefy śródręcza, którą określa okrąg śródręcza. Aby odnaleźć ten punkt należy przemieszczać się wzdłuż wykrytego palca aż do momentu odnalezienia pary punktów, czyli czubka palca *T* (ang. finger tip), oraz nasady *B* (ang. finger base). Te założenia zilustrowano na Rys. 30.



Rys. 29 Podział stref klasyfikacji wykrytych punktów charakterystycznych.
Czerwony – Strefa odrzutu punktów zbyt blisko konturu;
Zielony/niebieski – Podział na regiony dłoni i przedramienia poprzez linię nadgarstka;
Jasny kolor – Punkty klasyfikowane jako palce wystające;
Ciemny kolor – Punkty klasyfikowane jako palce zgięte;
Koła symbolizują strefy śródręcza i powiększone strefy śródręcza.

Najpierw analizowany jest region wykrytego punktu czubka palca *T1* w obrazie transformaty odległościowej z obrazu kierunkowego. Na obrazie transformaty odległościowej, na współrzędnych wykrytego punktu *T1* i o promieniu równym wartości transformaty w tym punkcie, czyli $r = DT_{P(x_P,y_P)}$ nanoszone jest koło z wartością 0 kasujące wszelkie potencjalne maksima. Następnie, w promieniu $r_2 = DT_{P(x_P,y_P)} \cdot \lambda$, szukane jest lokalne maksimum w punkcie *B1*. Jako że punkt ten nie może być wykryty w najbliższym otoczeniu punktu czubka palca, zostaje on wykryty w bliskiej odległości, na środku palca.

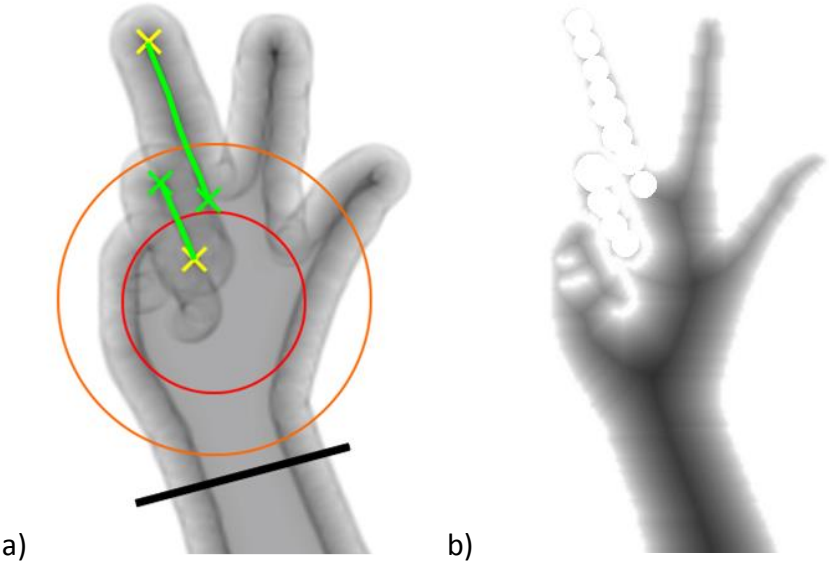
Po wykryciu tego punktu realizowana jest raz jeszcze ta sama procedura. Punkt oraz jego najbliższe sąsiedztwo jest wymazywane, a następnie odnajdywany jest kolejny punkt maksimum podstawiany pod *B1*. W ten sposób, wędrując po lokalnych maksimach umieszczonych wzdłuż palca, ostatecznie odnajdywany jest taki punkt, który znajduje się wewnątrz okręgu śródręcza. Wcześniej wykryty punkt bazowy *B1* staje się ostatecznym punktem nasady palca. Oznacza to, że punkt *B1* jest blisko okręgu śródręcza, jednocześnie poza nim, czyli $|SB1| \approx r_S$.Ten etap ilustruje Rys. 30 oraz algorytm z Rys. 34.



a) b)

Rys. 30 Przykład wykrycia punktu czubka palca wyprostowanego (żółty krzyżyk) i punktu nasady palca (zielony krzyżyk). a) Punkt wykryty jest na Rys. 25, b) Rys. 26-a służy do wykrycia ścieżki palca.

W przypadku, gdy kolejny punkt czubka palca T2 zostanie wykryty i sklasyfikowany jako zgięty palec, procedura podążania po maksimach transformaty odległościowej jest podobna, jednak kryterium stopu jest inne. Wyróżniane są dwa przypadki, czyli dwa kryteria stopu dla zgiętych palców. Pierwszy przypadek występuje, gdy czubek palca zostanie wykryty wewnątrz obszaru zamkniętego.

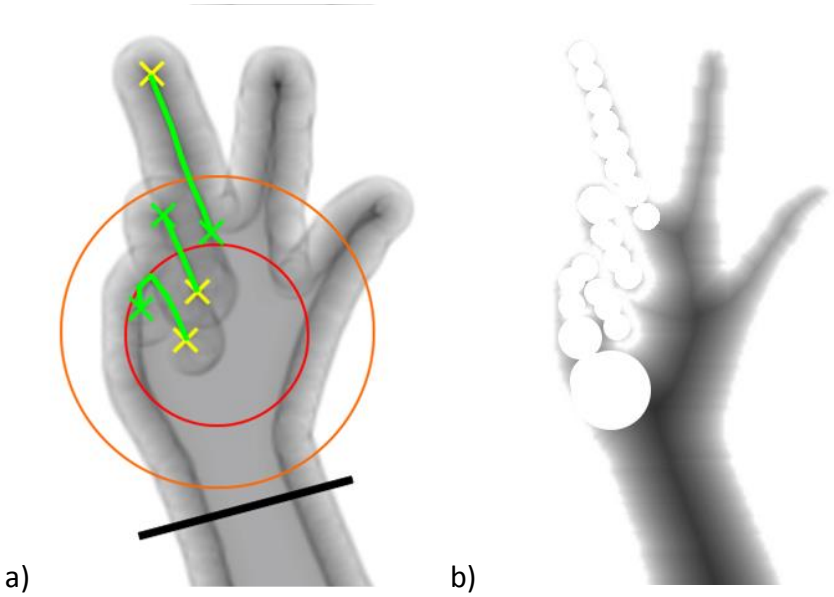


a) b)

Rys. 31 Przykład wykrycia punktu palca zgiętego (żółty krzyżyk) i punktu nasady palca (zielony krzyżyk), która aproksymowana jest na końcu danego regionu. a) Punkt wykryty jest na Rys. 25, b) Rys. 26-a służy do wykrycia ścieżki palca.

Obszar zamknięty to taki, w którym obraz kierunkowy tworzy zamknięty kontur, a lokalne maksima transformaty odległościowej otoczone są ze wszystkich stron wartościami 0. Region zamknięty można zaobserwować na Rys. 30 w obszarze palca serdecznego. Rys. 31 przedstawia procedurę odnalezienia nasady palca, w miejscu odnalezienia ostatniego lokalnego maksimum w danym rejonie.

W przypadku wykrycia punktu czubka palca *T3*, wewnątrz obszaru, który nie jest zamknięty, poszukiwanie kolejnych maksimów transformaty odległościowej wzdłuż palca mogłoby powodować wymazywanie regionów, które powinny być w przyszłości sklasyfikowane jako nowe punkty charakterystyczne. Dlatego wyznaczono statystycznie średnią długość palca zgiętego, która w przybliżeniu równa jest promieniowi regionu śródręcza. Kryterium stopu dla tego przypadku jest osiągnięte gdy punkt nasady palca oddali się od punktu czubka palca o odległość r_S. Wykrywanie palców zgiętych realizowane jest zgodnie z algorytmem przedstawionym na Rys. 35.

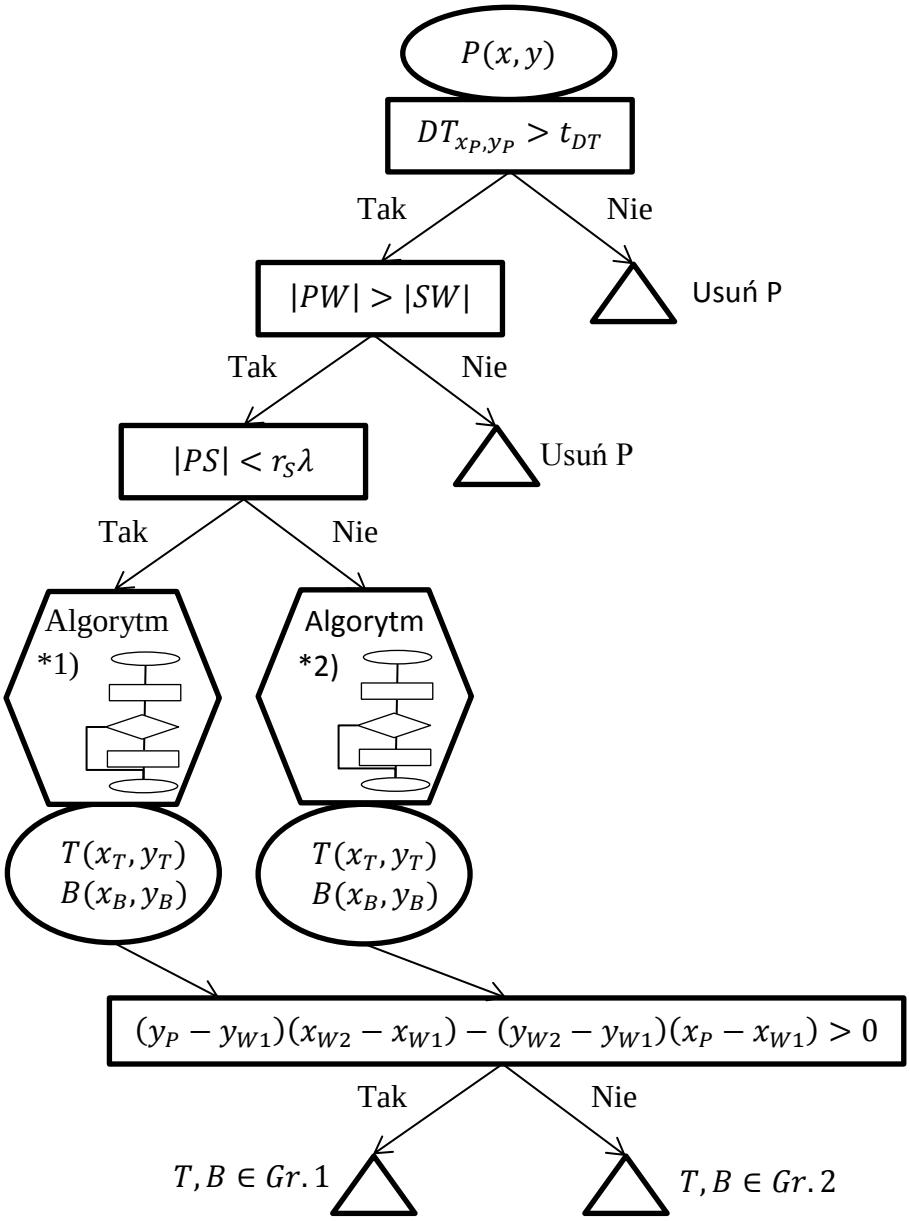


a) b)

Rys. 32 Przykład wykrycia punktu palca zgiętego (żółty krzyżyk) i wnioskowania nasady palca (zielony krzyżyk), która aproksymowana jest na postawie odległości. a) Punkt wykryty jest na Rys. 25, b) Rys. 26-a służy do wykrycia ścieżki palca.

Procedura wykrywania punktów charakterystycznych realizowana jest tak długo aż zostanie wykryta zadeklarowana wcześniej odpowiednia liczba punktów, lub gdy wykrycie nowych nie będzie możliwe. Wymazywanie przeanalizowanych już danych nanoszone jest również na obraz źródłowy, aby w miejscu wykrycia palców nie wykrywać nowych punktów charakterystycznych. Każda para punktów klasyfikowana jest do odpowiadającej sobie grupy, w zależności od tego, po której stronie nadgarstka została wykryta. Obserwacje wskazały, że region dłoni jest mniej jednorodny, co sprawia, że więcej punktów jest wykrywanych po stronie dłoni niż po stronie przedramienia. Dlatego ostatecznie ta grupa, która jest liczniejsza, klasyfikowana jest jako grupa zawierająca punkty charakterystyczne dłoni. Pozostała grupa jest odrzucana jako zbiór błędnie wykrytych punktów w regionie przedramienia.

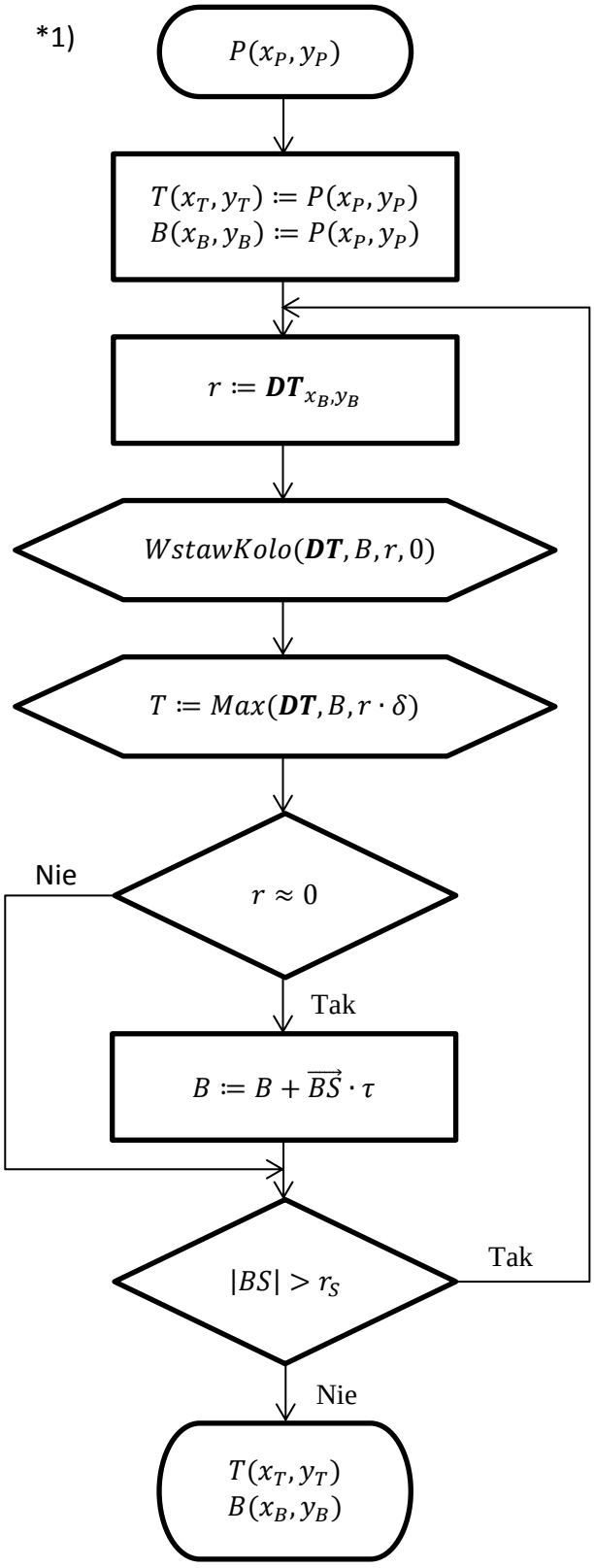
Podsumowując zbiór reguł można utworzyć drzewo decyzyjne przedstawione na Rys. 33.



Rys. 33 Drzewo decyzyjne klasyfikacji wykrytych punktów charakterystycznych

W zależności od tego, czy punkt wykryty został wewnątrz strefy śródręcza, czy poza nią, realizowany jest algorytm *1 przedstawiony na Rys. 34 lub *2 przedstawiony na Rys. 35. Obie metody różnią się w sposobie odnajdywania pary punktów T i B z wykrytego punktu P. Ostateczne przypisanie punktów do grup odbywa się na podstawie tego samego kryterium.

Podsumowując, celem przedstawionej metody jest wykrycie pary punktów T i B za pomocą odnalezionego punktu ekstremum P lub porzucenie tego punktu. W przypadku wykrycia punktów T i B dodatkowym celem jest przypisanie ich do grupy 1 lub grupy 2. Po wykryciu F_N punktów, mniej liczna grupa zostaje usunięta.



// Zakłada się punkt startowy P.

// Początkowo punkty T i B przyjmują
// współrzędne punktu P.

// Wartość promienia r pobierana jest
// z wartości transformaty
// odległościowej w punkcie B.

// Na obraz transformaty
// odległościowej DT, nanoszone jest
// koło w punkcie B, o promieniu r,
// i wartości 0

// Nowy punkt B odnajdywany jest jako
// maksimum w okręgu o promieniu *rσ,*
// środku T i wartościami z DT

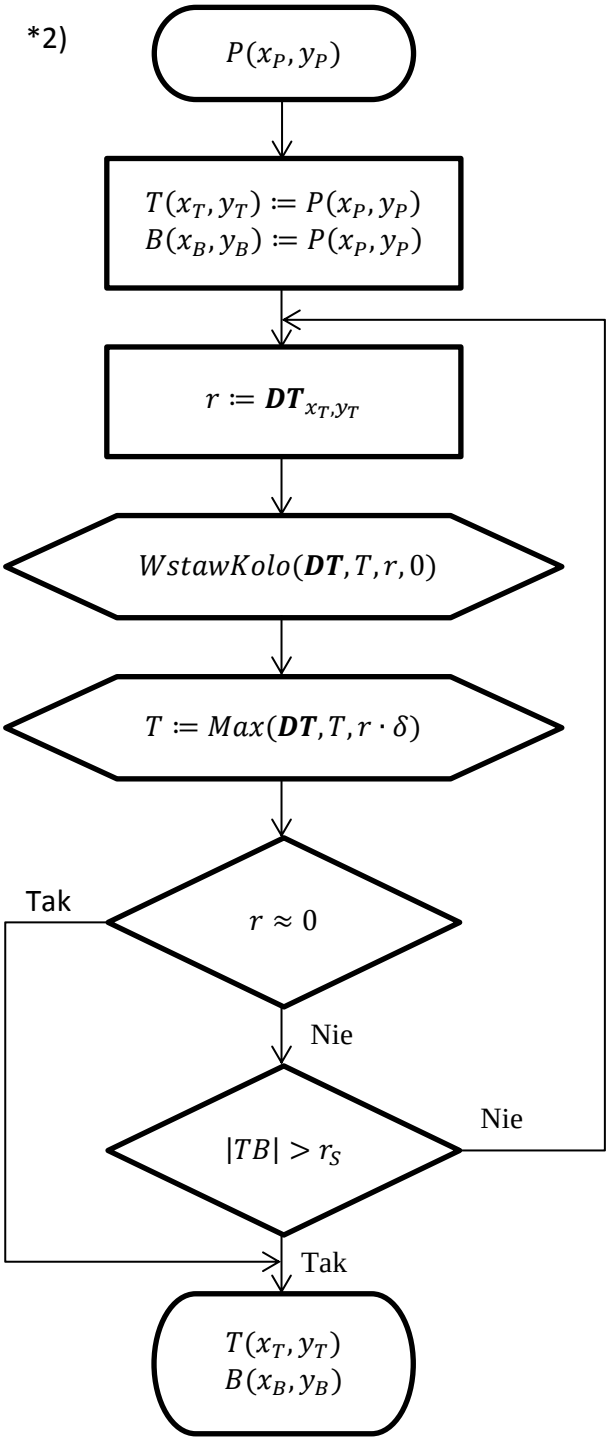
// Czy w regionie poszukiwań
// odnajdywane maksima dążą do 0?

// Następuje przesunięcie punktu B
// bliżej środka śródręcza S oraz wyjście
// z potencjalnego zapętlenia.

// Czy odległość pomiędzy punktami TB
// jest większa niż promień śródręcza?

// Koniec,
// odnaleziono parę punktów T i B

Rys. 34 Schemat blokowy algorytmu poszukiwania punktów palca wzniesionego.



// Zakłada się punkt startowy P.

// Początkowo punkty T i B przyjmują
// współrzędne punktu P.

// Wartość promienia r pobierana jest
// z wartości transformaty
// odległościowej w punkcie T.

// Na obraz transformaty
// odległościowej DT, nanoszone jest
// koło w punkcie T, o promieniu r,
// i wartości 0

// Nowy punkt T odnajdywany jest jako
// maksimum w okręgu o promieniu *rσ,*
// środku T i wartościami z DT

// Czy w regionie poszukiwań
// odnajdywane maksima dążą do 0?

// Czy odległość pomiędzy punktami TB
// jest większa niż promień śródręcza?

// Koniec,
// odnaleziono parę punktów T i B

Rys. 35 Schemat blokowy algorytmu poszukiwania punktów palca zdjętego

5. Walidacja eksperymentalna

Walidacja eksperymentalna ma na celu przeanalizowanie działania oraz jakości danych wyjściowych z prezentowanych algorytmów. Zbiór testów oraz ich wyników wykazuje prawidłowość tezy postawionej w niniejszej rozprawie doktorskiej.

Aby przeprowadzić serie testów należy posiadać obiekt badawczy. W tym celu została stworzona baza obrazów. Każdy obraz z bazy przedstawia gest wykonywany za pomocą odpowiedniego ułożenia dłoni, a każdy obraz dłoni posiada zbiór zaznaczonych punktów charakterystycznych. Celem algorytmów jest przeanalizowanie obrazu z bazy oraz wygenerowanie zbioru punktów charakterystycznych, jak najbardziej zbliżonych do tych z bazy.

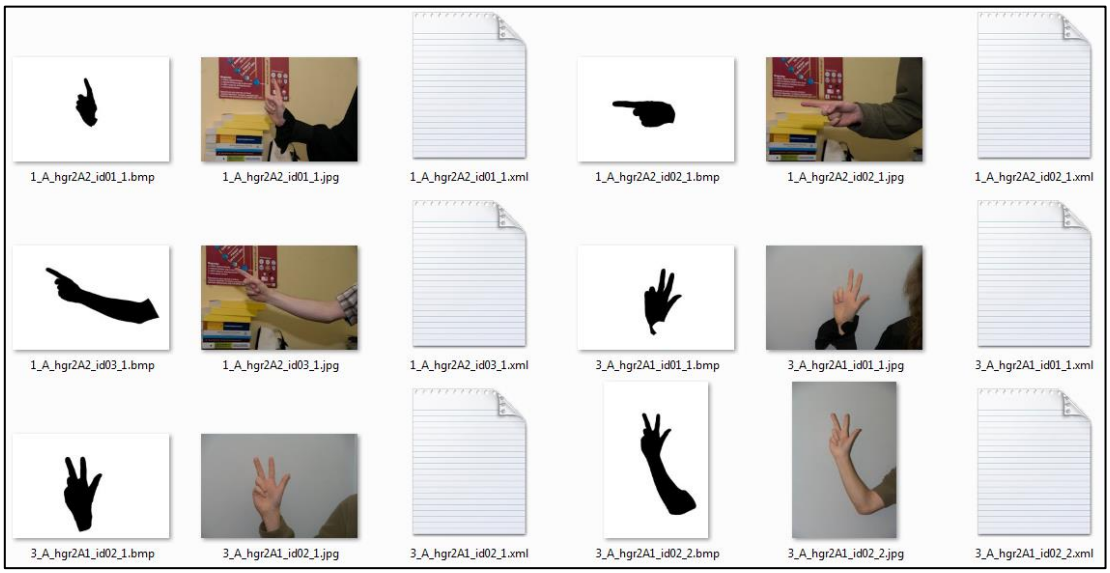
Przedstawione algorytmy przystosowane są do pracy na maskach wzorcowych zaznaczanych przez ekspertów, jak i na maskach, które są wynikami detekcji skóry. Jednak metody detekcji skóry nie są idealnie i niosą ze sobą błędy. Dlatego, we wszystkich prezentowanych wynikach, użyto masek wzorcowych, aby nie przenosić błędów generowanych na etapie segmentacji.

W celu określenia jakości wykrycia punktów przedstawiona została procedura walidacyjna. Procedura ta zawiera zbiór reguł, które klasyfikują wykryte punkty jako poprawnie wykryte lub błędnie wykryte.

Ostatecznie wyniki walidacji zostały przedstawione w formie wykresów oraz tabel, które przedstawiają dokładność oraz jakość wykrytych punktów charakterystycznych w obrazie. Wyniki te poddane są analizie, która prowadzi do wniosków końcowych.

5.1. Zbiór gestów i baza obrazów

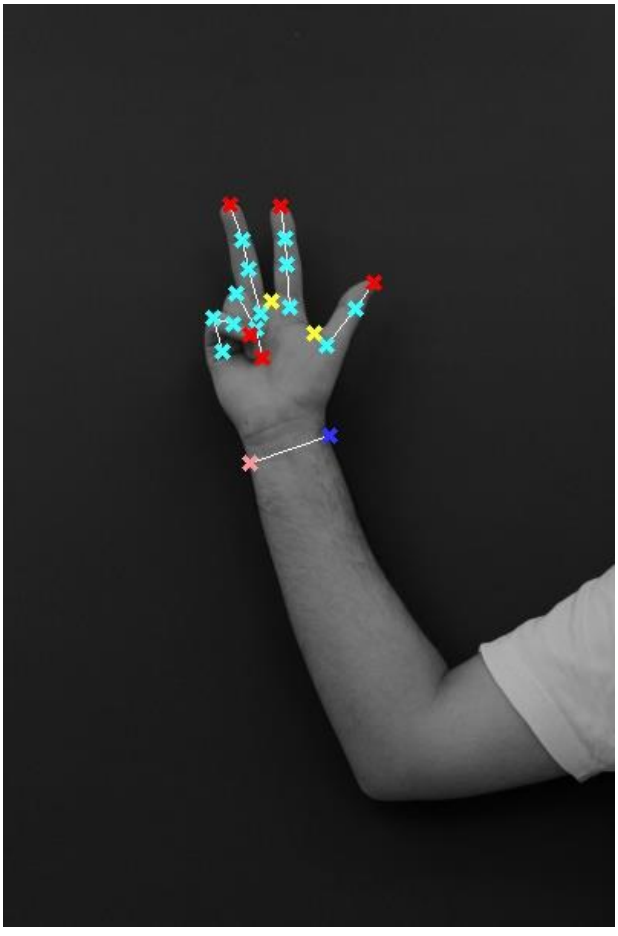
Na potrzeby przeprowadzenia serii testów oraz konieczności określenia jakości badanych algorytmów, powstała baza obrazów przedstawiająca zbiór gestów.



Rys. 36 Fragment bazy obrazów. Każdy obraz posiada maskę regionu skóry oraz plik XML z zaznaczonymi punktami charakterystycznymi przez eksperta.

```
<IMAGE WIDTH="3264" HEIGHT="4928" COLOR="TRUE">
  <hand>
    <FeaturePoint x="1734" y="2304" category="Wrist" type="WristThumb"/>
    <FeaturePoint x="1315" y="2444" category="Wrist" type="WristPinky"/>
    <FeaturePoint x="1720" y="1819" category="Thumb" type="Knuckle1"/>
    <FeaturePoint x="1879" y="1625" category="Thumb" type="Knuckle2"/>
    <FeaturePoint x="1976" y="1482" category="Thumb" type="Fingertip"/>
    <FeaturePoint x="1660" y="1755" category="Thumb" type="ConcavePoint"/>
    <FeaturePoint x="1522" y="1616" category="IndexFinger" type="Knuckle1"/>
    <FeaturePoint x="1507" y="1384" category="IndexFinger" type="Knuckle2"/>
    <FeaturePoint x="1502" y="1245" category="IndexFinger" type="Knuckle3"/>
    <FeaturePoint x="1479" y="1074" category="IndexFinger" type="Fingertip"/>
    <FeaturePoint x="1430" y="1582" category="IndexFinger" type="ConcavePoint"/>
    <FeaturePoint x="1369" y="1646" category="MiddleFinger" type="Knuckle1"/>
    <FeaturePoint x="1307" y="1409" category="MiddleFinger" type="Knuckle2"/>
    <FeaturePoint x="1269" y="1257" category="MiddleFinger" type="Knuckle3"/>
    <FeaturePoint x="1204" y="1066" category="MiddleFinger" type="Fingertip"/>
    <FeaturePoint x="1242" y="1544" category="RingFinger" type="Knuckle2"/>
    <FeaturePoint x="1343" y="1731" category="RingFinger" type="Knuckle3"/>
    <FeaturePoint x="1376" y="1883" category="RingFinger" type="Fingertip"/>
    <FeaturePoint x="1165" y="1850" category="Pinky" type="Knuckle1"/>
    <FeaturePoint x="1118" y="1674" category="Pinky" type="Knuckle2"/>
    <FeaturePoint x="1223" y="1703" category="Pinky" type="Knuckle3"/>
    <FeaturePoint x="1315" y="1759" category="Pinky" type="Fingertip"/>
  </hand>
  <hand/>
</IMAGE>
```

Rys. 37 Wnętrze pliku XML z opisem pozycji i typu punktów charakterystycznych dla Rys. 38.



Rys. 38 Wizualizacja oznaczonych punktów charakterystycznych, zapisanych w pliku XML z Rys. 37.

W skład bazy wchodzą 3 serie, każda zawierająca zdjęcia przedstawiające użytkowników prezentujących pewien zbiór gestów. Do każdego obrazu z bazy dołączona jest maska regionu skóry oraz wskazane położenie punktów charakterystycznych w pliku XML. Zarówno maska jak i zbiór punktów są danymi oznaczonymi przez ekspertów, czyli użytkowników, którzy ręcznie wskazywali te dane z użyciem specjalnego oprogramowania. Dane te traktowane są jako dane wzorcowe, czyli GT (ang. ground truth). Baza wykorzystywana do rozpoznawania gestów (ang. hand gesture recognition, hgr), składa się z:

- hgr1:
  Pierwsza baza służąca głównie do testów algorytmów detekcji skóry. Bazę tworzą zdjęcia 12 osób prezentujących 25 gestów głównie z Polskiego Języka Palcowego.
  - 899 obrazów o rozdzielczości do 640×480;
  - 12 osób;
  - 25 gestów;
  - Niekontrolowane tło oraz oświetlenie.
- hgr2a:
  W celu testowania algorytmów wykrywania punktów charakterystycznych powstał pomysł stworzenia bazy bardziej wymagającej oraz posiadającej obrazy o wyższej rozdzielczości. Ta baza była próbą przed stworzeniem bazy hgr2B.
  - 85 obrazów o rozdzielczości 4672×3104;
  - 3 osoby;
  - 13 gestów;
  - Kontrolowane tło i oświetlenie.
- hgr2b:
  W skład tej bazy wchodzi kilka gestów prostych do wykrycia, jednak głównie znajdują się w niej gesty skomplikowane oraz bardzo podobne, wybrane z Polskiego i Amerykańskiego Języka Palcowego.
  - 574 obrazów o rozdzielczości 4928×3264;
  - 18 osób;
  - 32 gestów;
  - Kontrolowane tło i oświetlenie.

Każdy obraz, oraz jego dane wzorcowe znakowane są za pomocą nazwy pików. Przykładowo, na Rys. 38 widać obraz o nazwie: 3_A_hgr2B_id01_1. Kolejno oznacza to:

- 3 – Obraz przedstawia gest rozumiany jako liczba „3”. Możliwe są oznaczenia znaków odpowiadających za litery, liczby lub znaki specjalne;
- A – Gest zawierający się w Amerykańskim Języku Palcowym. Możliwe są gesty z grupy Polskiego lub Amerykańskiego Języka Palcowego, oraz gesty specjalne;
- hgr2B – obraz należący do bazy hgr2B;
- Id01 – Osoba o identyfikatorze 01;
- 1 – Osoba id01 pokazała ten gest pierwszy raz.

Baza powstała w trakcie realizacji projektu badawczego IP2011 023071. Obrazy, maski oraz zbiory punktów charakterystycznych zapisanych w postaci plików XML, można znaleźć pod adresem internetowym http://sun.aei.polsl.pl/~mkawulok/gestures/ . Badania z użyciem tej bazy znajdują się między innymi w publikacjach [57, 58, 73, 76, 77]

5.2. Procedura walidacyjna

Procedura walidacyjna dobrana została w zależności od zestawu testowego, który podlegał badaniom. W kolejnych podrozdziałach przedstawiona została metoda oceny poszczególnych kroków algorytmów zarówno pod względem jakości dopasowania, jak i liczby wskazanych punktów.

Poza błędem lokalizacji wykrytych punktów, ważnym wskaźnikiem jakości algorytmu jest również jego czas wykonywania. W trakcie testów przebadano również średni czas analizy jednego obrazu. Wszystkie testy przeprowadzane były na komputerze Asus R500V K55VM o parametrach:

- Procesor Intel Core i7-3610QM (4 rdzenie, od 2.30 GHz, 6 MB cache Ivy Bridge)
- Chipset Intel HM76 Express
- Pamięć RAM 4096 MB (SO-DIMM DDR3, 1600 MHz)
- Karta graficzna NVIDIA GeForce GT 630M + Intel HD Graphics 4000

5.2.1. Ocena metod wykrywania nadgarstka

Jednym z podstawowych kroków służących ustaleniu regionu dłoni jest ustalenie miejsca nadgarstka. Dla części algorytmów, odnalezienie nadgarstka jest pierwszym krokiem metody poszukiwania punktów charakterystycznych [22, 74] a jego poprawna lokalizacja wpływa na lokalizacje pozostałych punktów charakterystycznych. Zatem algorytm poszukiwania nadgarstka został przebadany oddzielnie. W celu sprawdzenia jakości wykrytych punktów nadgarstka zaproponowano metodę walidacyjną polegającą na porównaniu różnicy współrzędnych wykrytego punktu nadgarstka oraz pobranego z bazy. Punkt nadgarstka *W* zlokalizowany jest w połowie pomiędzy punktami brzegowymi.

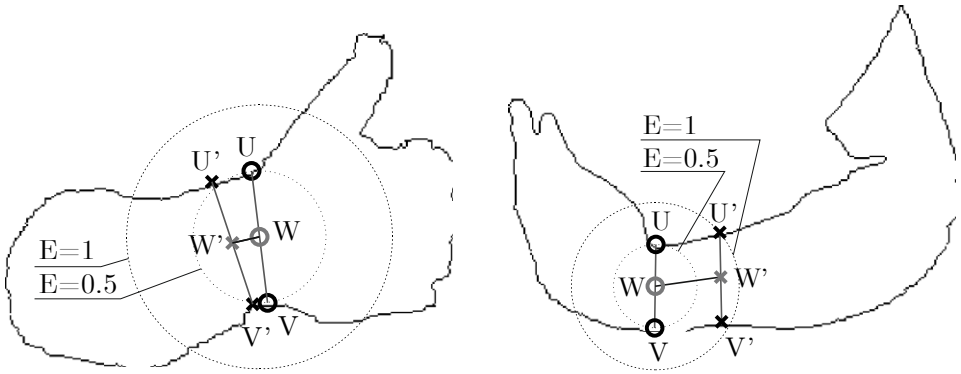
Oznaczenie punktu nadgarstka jest wymagane w programie do oznaczania obrazów, zaś większość algorytmów poszukujących nadgarstek nie zakłada jego nieodnalezienia, zatem zbiór testowy zawsze będzie posiadał pary punktów wzorcowych *W*, oraz punktów wykrytych *W'*. Zakłada się więc, że jeżeli w obrazie nie wykryto lub nie oznaczono nadgarstka, wtedy dłoń na obrazie nie występuje.

Procedurę walidacyjną stanowi porównanie wzajemnej różnicy pomiędzy punktem wykrytym *W'* oraz wzorcowym *W*. Ich wzajemna odległość, określa przesunięcie punktu wykrytego od pożądanego, czyli innym słowy jest to błąd dopasowania. Błąd dopasowania wyrażany jest w pikselach. Jako, że wielkość ta zależna jest od rozmiarów obrazu wejściowego, należy wartość tę znormalizować. Zakładając, że dłoń jest proporcjonalnie stała z niewielką różnicą pomiędzy osobami [94], w celu normalizacji błędu dopasowania wystarczy podzielić wyznaczony błąd przez znaną odległość, jak na przykład promień regionu

śródręcza, grubość palca lub szerokość nadgarstka. Z bazy wzorcowej najłatwiej uzyskać dwa punkty nadgarstka, każda wartość błędu dopasowania normalizowana jest przez podzielenie jej przez odległość między punktami nadgarstka, czyli grubość nadgarstka. Podsumowując, błąd wykrycia nadgarstka liczony jest jako

$$e = \frac{|WW'|}{|UV|}. \tag{20}$$

Przedstawiany w dalszych badaniach błąd będzie miał wartości zaczynające się od 0. Dla wartości błędu równego 0, punkt wykryty został dokładnie w miejscu, w którym został on wskazany przez eksperta w bazie. Jeżeli błąd będzie miał wartość 0,1 oznacza to, że punkt został wykryty w odległości równej 10% grubości nadgarstka.



Rys. 39 Przykłady liczenia błędu dopasowania wykrytych punktów nadgarstka U', V' i W' do punktów wzorcowych U, V i W. Okręgi o promieniu E pokazują granicę znormalizowanego błędu.

Powstaje jednak zasadnicze pytanie: Jaka wartość błędu jest granicą między poprawnym wykryciem, a błędnym. W tym celu należy ustalić wartość graniczną błędu *E*, która odgraniczać będzie punkty źle wykryte od punktów wykrytych poprawnie. Wartość błędu granicznego *E* oraz sposób wyznaczenia opisano w rozdziale 5.3.

5.2.2. Ocena metod wykrywania pozostałych punktów charakterystycznych

Po przebadaniu metod poszukiwania nadgarstka i po zaimplementowaniu dalszych metod poszukiwania pozostałych punktów charakterystycznych, należało odpowiednio zmodyfikować procedurę oceniania wyników dopasowania. W przypadku nadgarstka wystarczyło porównać dwa punkty ze sobą, wykryty i dopasowany. Zakładano bowiem, że oba punkty istnieją.

Aby przebadać algorytmy wykrywania punktów charakterystycznych, stworzono procedurę walidacyjną mającą na celu określenie dokładności lokalizacji wykrytych punktów charakterystycznych. Danymi testowymi są zbiór punktów wzorcowych pobranych z bazy oraz zbiór punktów wykrytych za pomocą testowanego algorytmu. Należy zauważyć, że liczba punktów wykrytych i wzorcowych nie musi być sobie równa. Dla każdego punktu wykrytego liczona jest odległość do każdego punktu wzorcowego. Wyniki umieszczane są w

macierzy wzajemnej odległości punktów. Wymiary macierzy *n*×*m,* zależne są od liczby punktów, gdzie wymiar kolumn *m* równy jest liczbie punktów wzorcowych, a wymiar wierszy *n* równy liczbie punktów wykrytych.

Po ustaleniu macierzy wzajemnej odległości, punkty dobierane są w pary. Jeden wykryty punkt jest sparowany z jednym punktem wzorcowym. Taka procedura jest wymagana, ponieważ często algorytmy przypisują punkty jedynie do grupy, takiej jak czubek palca lub nasada palca, jednak nie zawsze identyfikują je jako konkretny palec. Procedura walidacyjna została zaprojektowana w taki sposób, aby porównywać między sobą punkty konkretnej grupy, które nie posiadają identyfikatora konkretnego palca. To znaczy, że otwarta dłoń z widocznymi pięcioma palcami będzie miała 5 punktów czubków palców wzorcowych oznaczonych przez ekspertów oraz najprawdopodobniej 5 punktów czubków palców wykrytych. Jako, że nie wiadomo który z punktów należy do którego palca, celem opisywanej procedury jest sparowanie pięciu punktów wykrytych z pięcioma punktami wzorcowymi.

Zgodnie z przykładem z Tab. 3 a, punkty sparowane zostaną w następujący sposób: P1-D1, P2-D2, P5-D3, P4-D4. Punkt P3 nie będzie posiadał pary. Analizując macierz wzajemnej odległości, szukana jest pozycja minimum. Numer wiersza i kolumny odpowiada punktom wzorcowym i wykrytym, które będą sparowane. Ze względu na to, że te punkty mają już parę i nie mogą mieć kolejnej pary, zostają one wykasowane z macierzy. Po sparowaniu punktu, wiadomo że wykryty punkt został uznany jako poprawnie wykryty z nieznormalizowanym błędem równym odległości od punktu wzorcowego, wyrażonego w pikselach. Punkty są parowane aż do momentu, gdy jeden z wymiarów macierzy równy jest 0. Może zaistnieć sytuacja, w której liczba wierszy i kolumn nie jest sobie równa. Oznacza to, że na etapie detekcji wykryto zbyt dużo punktów lub zbyt mało punktów.

Tab. 3 Przykład macierzy wzajemnej odległości punktów wykrytych D do punktów wzorcowych P. Dobór par punktów zależny jest od wartości odległości pomiędzy punktami D i P, zwizualizowanych natężeniem szarości pól. Kolejne tabele przedstawiają różne metody parowania punktów. a) Wybór kolejnych wartości minimów globalnych, b) wybór tych pól, których suma jest najmniejsza, c) dopasowanie błędne, niezgodne z założeniami.

a)	P1	P2	P3	P4	P5
D1	5	9	77	85	79
D2	7	13	78	91	90
D3	59	70	33	18	14
D4	86	95	25	20	60

Suma: 52

b)	P1	P2	P3	P4	P5
D1	5	9	77	85	79
D2	7	13	78	91	90
D3			33	18	14
D4			25	20	60

Suma: 50

c)	P1	P2	P3	P4	P5
D1	5	9	77	85	79
D2	7	13	78	91	90
D3	59	70	33	18	14
D4	86	95	25	20	60

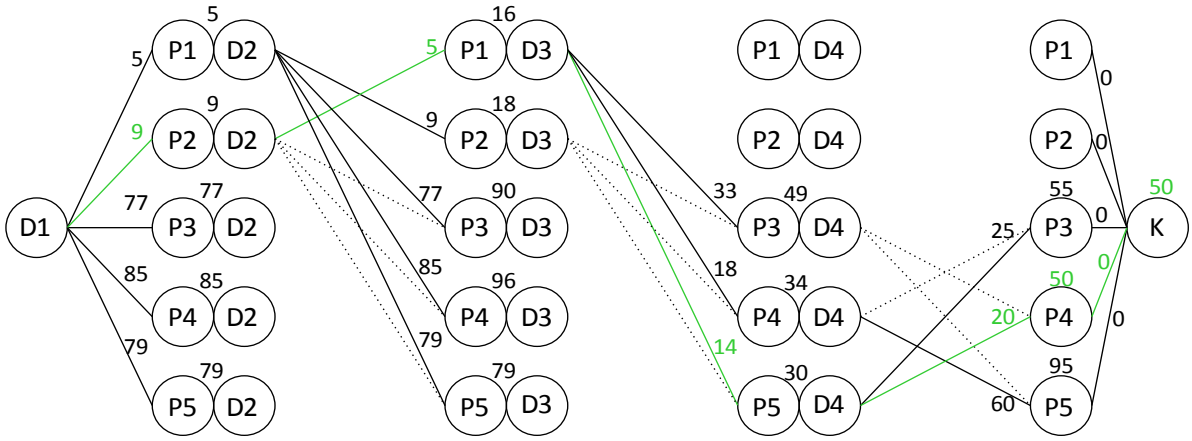
Suma: 44

Rozwiązanie zaprezentowane w przykładzie z Tab. 3 a, nie jest optymalne. Licząc sumę wszystkich odległości punktów sparowanych, otrzymujemy sumę 52. Istnieje jednak możliwość takiego sparowania punktów, że ich suma odległości będzie niższa. Pokazuje to przykład Tab. 3 b. Jeżeli punkty dobrane zostaną w pary: D1-P2, D2-P1, D3-P5, D4-P4, suma ich odległości będzie wynosić 50. Ostatecznie, zatem, postanowiono zmodyfikować procedurę dobierania par punktów, przeszukując wszystkie możliwe kombinacje, oraz odnajdując takie pary, dla których suma ich odległości jest najniższa.

Należy jednak pamiętać o założeniu, że punkty raz sparowane, nie mogą występować w innej parze. Przykład z Tab. 3 c, nie spełnia tego założenia. Przykładowo, algorytm

wykrywający czubek palca wskazującego powinien wykryć tylko jeden punkt, który można przypisać do tej kategorii. Podobnie, punkt wykryty pomiędzy palcem środkowym a wskazującym nie może jednocześnie być przypisany do obu tych palców.

Dodatkowo, w metodzie poszukiwania zaimplementowano algorytm Dijkstry. Rozwiązanie problemu parowania punktów przedstawia graf na rysunku Rys. 40. Ze względu na to, że punktów D jest mniej niż punktów P, każdy punkt D powinien mieć parę. Dlatego graf skonstruowany jest w taki sposób, że dla kolejnych punktów D, sprawdzana jest odległość do każdego możliwego punktu P. Odległości między konkretnymi punktami P i D nie zmieniają się, dlatego koszty każdej krawędzi, która połączona jest z danym punktem P, są identyczne. Po odnalezieniu pary dla punktu D1, poszukiwana jest para dla punktu D2. Punkt D2 nie może być połączony z takimi punktami P, które były wykorzystywane we wcześniejszych operacjach. Skoro punkty D parowane są z punktami P, to każdy wierzchołek, poza startowym, jest reprezentowany jako odpowiedni punkt P oraz odpowiedni punkt D. Celem algorytmu Dijkstry jest odnalezienie najkrótszej ścieżki w prezentowanym grafie, która łączy wierzchołek D1 z wierzchołkiem końcowym K. W trakcie wykonywania algorytmu niektóre odległości miedzy punktami, które nie będą parami, nie są obliczane. Odległości, które nie były wymagane do wyznaczenia poprawnego wyniku reprezentują puste pola w Tab. 3 b oraz niepołączone węzły w grafie.



Rys. 40 Graf prezentujący przykład rozwiązania problemu przedstawionego w Tab. 3 z użyciem algorytmu Dijkstry. Liniami przerywanymi oznaczono krawędzie, które poddano relaksacji. Optymalne rozwiązanie przedstawia kolor zielony.

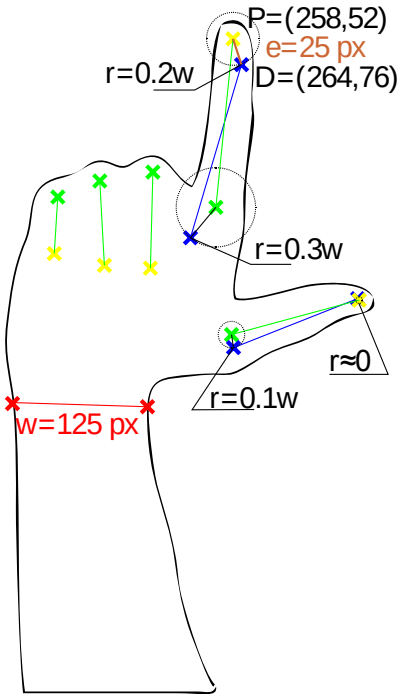
Powyższa procedura jest przeprowadzana dwukrotnie, dla grupy czubków palców oraz dla grupy nasady, ponieważ algorytmy wykrywające nasady palców rozgraniczają grupy punktów. W przypadku punktów nadgarstka, metoda ta również może być zastosowana, jednak wymiary macierzy będą *1×1*, co sprowadzać ją będzie do metody opisanej w poprzednim rozdziale.

Po przeanalizowaniu macierzy odległości, każdy z wykrytych punktów posiada błąd dopasowania. Wyjątkiem są nadmiarowo wykryte punkty, które nie są sparowane.

Na przykładzie z Rys. 41 zaznaczono niebieskimi krzyżykami cztery wykryte punkty charakterystyczne. Dwa z nich mają etykietę czubka palca, a dwa nasady palca. Punkty te porównano z pięcioma wzorcowymi punktami nasady i pięcioma punktami czubków palców. Można zaobserwować, że jeden z punktów wykryty został w tych samych współrzędnych, w

których wskazał go użytkownik. Jego błąd dopasowania równy jest 0. Do punktu *D*, zaznaczonego na rysunku sparowano punkt *P.* Ich wzajemna odległość wynosi 25 pikseli. Szerokość nadgarstka wzorcowego wynosi *w=124* piksele. W tym przypadku, błąd dopasowania wynosi około *e=0,2.* Można wyobrazić sobie, że jeżeli punkt zostałby wykryty gdziekolwiek na okręgu o promieniu *r=0,2w,* jego błąd wciąż byłby równy *e=0,2.*

Dla każdego obrazu z bazy danych, po etapie walidacji, uzyskiwany jest zbiór błędów dopasowania każdego odnalezionego punktu.



Rys. 41 Przykład liczenia błędu dopasowania dla sparowanych punktów wykrytych (niebieskie) oraz punktów wzorcowych (zielone – nasady, żółte – czubki palców). Błąd dopasowania normalizowany jest przez szerokość nadgarstka wzorcowego (czerwony)

Ostatecznie w celu ustalenia dokładności działania algorytmu punkty klasyfikowane są w następujący sposób:

- Poprawne – Punkty wykryte, których wartość błędu jest mniejsza od błędu granicznego, czyli *e≤E;*
- Błędne – Punkty wykryte, których wartość błędu jest większa od błędu granicznego, czyli *e>E*;
- Nadmiarowe – Punkty nadmiarowo wykryte, które nie zostały sparowane z żadnym punktem wzorcowym. Występują gdy *n>m*. Przykładem może być wykrycie pięciu palców w przypadku gdy obraz przedstawia zaciśniętą pięść z niewidocznymi czubkami palców. Liczba punktów nadmiarowych dla danego obrazu wynosi *n-m*;
- Niewykryte – Punkty, które nie zostały wykryte. Jeżeli istnieją punkty wzorcowe, które nie posiadają pary, oznacza to, że nie wykryto istotnych punktów. Występują gdy *n<m*. Przykładem może być nie wykrycie żadnych punktów na dłoni z widocznymi palcami. Liczba punktów dla obrazu wynosi *m-n*;

W standardowych miarach oceny klasyfikatorów używana jest nomenklatura klasyfikująca poprawne próby jako prawidłowe wskazanie (ang. TP – true positive) lub prawidłowe niewskazanie (ang. TN – true negative). Porażkę opisuje się jako niepoprawne wskazanie (ang. FP – false positive) lub niewskazanie poprawnego elementu (ang. FN – false negative). Korzystając z tych miar można wyznaczyć wartość predykcyjną dodatnią, nazywaną precyzją oraz czułość nazywaną selektywnością.

Precyzja PPV określana jest przez stosunek wskazań prawdziwie dodatnich oraz sumy prawdziwie dodatnich i fałszywie dodatnich

$$PPV = \frac{TP}{TP + FP}. \tag{21}$$

Selektywność TPR określana jest przez stosunek wskazań prawdziwie dodatnich oraz sumę prawdziwie dodatnich i fałszywie ujemnych

$$TPR = \frac{TP}{TP + FN}. \tag{22}$$

Miarą testu jest wynik F1, czyli miara średniej harmonicznej precyzji i selektywności

$$F1 = \frac{2TP}{2TP + FN + FP}. \tag{23}$$

W przypadku wykrywania punktów charakterystycznych można dokonać następującej klasyfikacji. Punkty poprawne klasyfikowane są jako TP, a punkty błędne klasyfikowane są jako FP. Granicę między TP a FP określa wartość błędu granicznego E. Punkty niewykryte traktowane są jako FN. Ostatecznie, punkty nadmiarowe, czyli takie którym nie odnaleziono pary, klasyfikowane są również jako FN. Pojawia się jednak problem dotyczący punktów TN. Zgodnie z definicją, każdy punkt na obrazie, który nie został uznany jako punkt charakterystyczny uznawany jest jako TN. Jeżeli dla danego obrazu, punktów TP, FP i FN może być kilkanaście, to punktów TN byłoby kilka tysięcy lub milionów. Dlatego test opierający się na TN nie byłby miarodajny.

Z powodu braku punktów prawdziwie ujemnych TN, nie ma możliwości zastosowania innych miar statystycznych pozwalających na ocenę testu. Jednak zbiór błędów można oceniać jeszcze za pomocą histogramów oraz wykresów dystrybuanty. Analizując cały test, dla zadanej liczby obrazów uzyskiwany jest zbiór błędów dopasowania punktów wykrytych do wzorcowych. Posortowana i zsumowana liczebność kumulacji błędu w danym przedziale może posłużyć wizualizacji rozkładu błędu danej metody, oraz w przypadku naniesienia wielu histogramów lub dystrybuant na jeden wykres – porównaniu kilku metod. Wykresy te stanowią dobrą wizualizację jakości działania algorytmu, jednak nie przenoszą bezpośredniej informacji o skuteczności testu, w przeciwieństwie do opisanych wcześniej miar

statystycznych. Wyjątkiem jest opracowana metoda wyznaczania parametrów sterowania algorytmem.

5.2.3. Metoda wyznaczania parametrów sterujących

Procedura walidacyjna została zmodyfikowana tak aby stworzyć algorytm uczący się, mający na celu dostrojenie parametrów proponowanego algorytmu, bazującego na obrazie kierunkowym. Dystrybuantę błędu łatwo wyznaczyć po teście, jako zliczenie liczby występowania błędów z danego zakresu. Zakłada się zatem, że jakość algorytmu zależna jest od pola powierzchni pod wykresem dystrybuanty. Pożądane jest, aby wszystkie wykryte punkty miały błąd *e=0*. W takim przypadku oznaczać to będzie, że wszystkie punkty zostały wykryte idealnie, a wykres dystrybuanty będzie wyglądał jak skok jednostkowy.

W celu poprawienia jakości wykrycia zaproponowano funkcje celu *I*, którą należy maksymalizować. Funkcja ta jest wyznaczona jako pole powierzchni pod wykresem dystrybuanty, czyli całka z funkcji dystrybuanty w przedziale błędów dopasowania. Całka ta może zostać uproszczona i przedstawiona jako całka Riemanna, czyli jako suma wszystkich wartości dystrybuanty przemnożonych przez długość przedziału Δ_i. Jako, że długość przedziału jest stała, ponieważ wykres dystrybuanty liczony jest ze stałym krokiem, można dokonać założenia, że $\Delta_i = \frac{1}{N}$, gdzie N to liczba przedziałów błędu. Po wyliczeniu wykresu dystrybuanty, dla N przedziałów błędu, funkcja kosztu liczona jest jako suma wszystkich N wartości z wykresu, podzielona przez liczbę przedziałów N.

$$I = \int_0^n F(x)dx = \sum_{i=1}^{N} F(t_i)\Delta_i = \sum_{i=1}^{N} \frac{F_i}{N} \qquad (24)$$

5.3. Analiza wyników

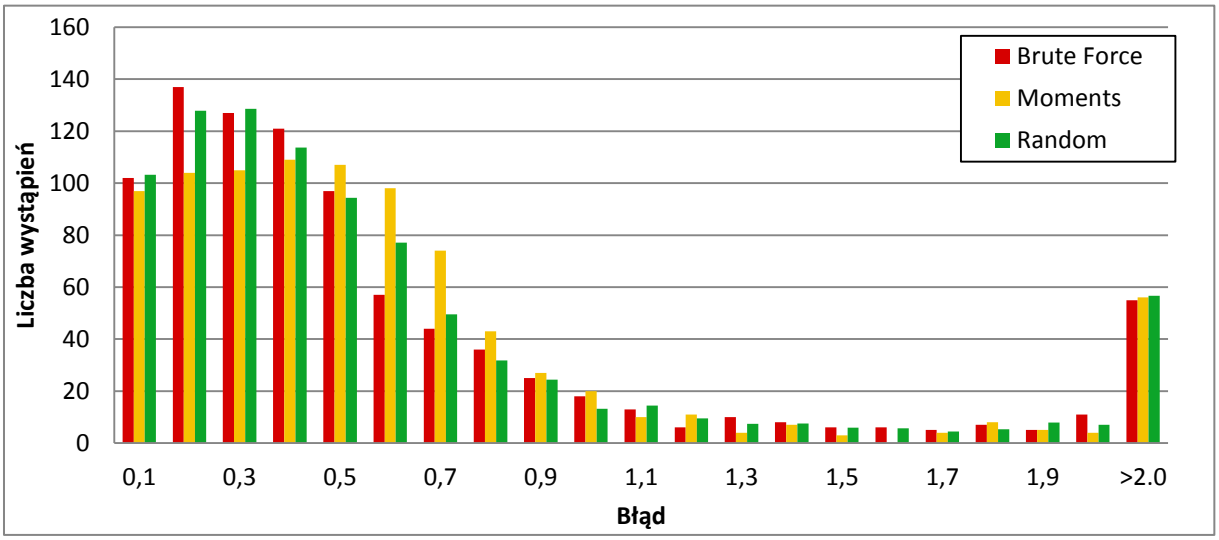
Zgodnie z procedurami walidacyjnymi opisanymi w poprzednim rozdziale, przeprowadzono serie testów. Dla zbioru testowego z bazy obrazów, przeprowadzono test seryjny algorytmów zaimplementowanych we wspólnym środowisku programistycznym. Algorytmy wykonywane były kolejno, dla tego samego zbioru obrazów, a następnie uzyskany zbiór punktów charakterystycznych zapisanych w postaci pliku XML, porównywany był ze zbiorem punktów oznaczonych przez eksperta. Otrzymane wyniki zostały opisane w odpowiednich podrozdziałach niniejszego rozdziału.

5.3.1. Badanie metod poszukiwania punktów nadgarstka

Pierwszym przeprowadzonym testem było sprawdzenie jakości działania algorytmu poszukiwania nadgarstka. Zaproponowany algorytm, opisany w rozdziale 4.1, wykorzystywał zliczenie profilu obrazu obróconego o kąt wyznaczony przez najdłuższą przekątną profilu. Do testów wybrano następujące metody:

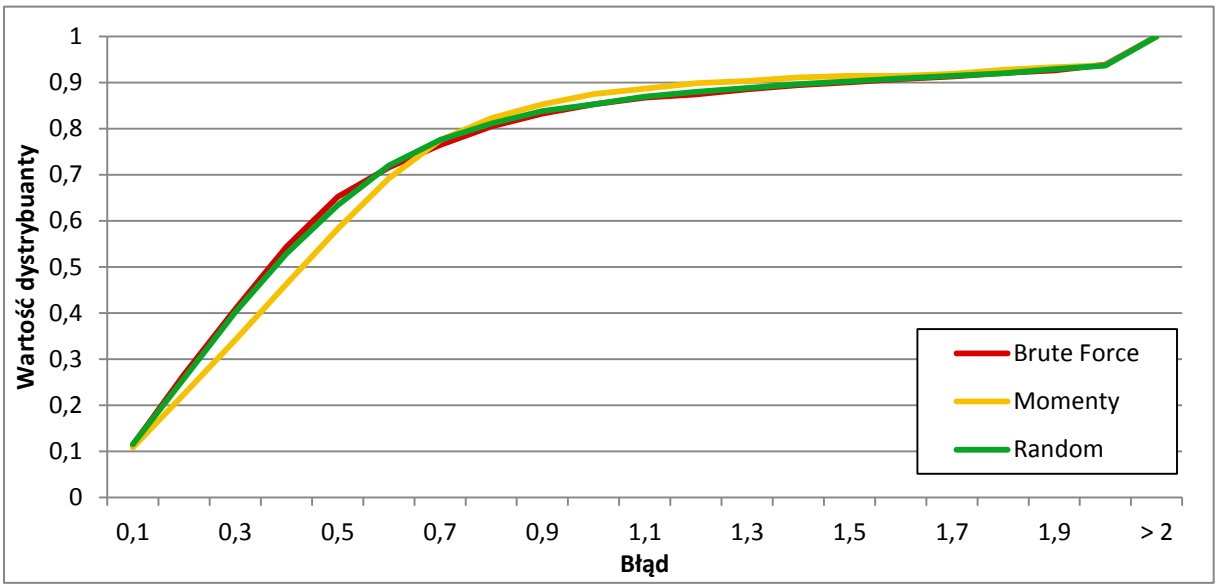
1. Random – Zaproponowana metoda wybierania najdłuższej przekątnej spośród zbioru losowo wybranych cięciw konturu.
2. Brute Force – Metoda poszukiwania dokładnie najdłuższej przekątnej.
3. Moments – Metoda wykorzystująca wyznaczenie momentów obrazu [75].

Dokładny opis metod można odnaleźć w rozdziale 4.1. Po przeprowadzeniu pierwszego testu na bazie hgr1 otrzymano następujące wyniki:



Rys. 42 Histogram błędu dopasowania punktu nadgarstka liczony dla 3 metod.

Zarówno wykres histogramu jak i wykres dystrybuanty wskazują, że najlepszą metodą jest odnajdywanie nadgarstka metodą siłową, czyli Brute Force. Metoda Random nie pozostaje jednak znacząco w tyle. Oznacza to, że zaproponowana metoda wyznaczania najdłuższej cięciwy liczy punkty nadgarstka z większą precyzją niż metoda wykorzystująca momenty. Dodatkowo, zaproponowana optymalizacja czasu wykonywania algorytmu w znaczący sposób skróciła ten czas, mając niewielki wpływ na błąd wyniku końcowego. Wyniki testów można zaobserwować w Tab. 4.



Rys. 43 Dystrybuanta błędu dopasowania punktu nadgarstka liczona dla 3 metod.

Dodatkowo, dla każdej metody zbadano czas potrzebny do uzyskania wyniku z każdego obrazu. Testy pokazały, że przeszukiwanie wszystkich możliwych kombinacji par punktów konturu zajmuje średnio niecałą minutę. Głównym powodem tak długiego czasu jest sprawdzenie, dla każdej pary, warunku zawierania się cięciwy wewnątrz konturu. Taka metoda nie mogłaby być wykorzystywana w docelowym programie, jednak została zaimplementowana, aby pokazać maksymalne możliwości algorytmu.

Tab. 4 Wyniki porównania testów algorytmów odnajdywania nadgarstka

		Random	Brute Force	Moments
Czas wykonywania dla 1 obrazka	$t_{śr}$ [s]	0,0812±0,0322	49,680±27,276	0,0254±0,0032
	t_{max} [s]	0,359	176,111	0,042
Błąd całkowity	$e_{śr}$	0,5940±0,6791	0,5937±0,6832	0,6016±0,6563
Wyniki dla błędu granicznego E=1,0	Suma e>E	131,7(14,7%)	132(14,7%)	111(12,4%)
	$e_{śr}$>E	0,3005	0,2990	0,3439
Wyniki dla błędu granicznego E=0,5	Suma e>E	322,3(36,0%)	309(34,4%)	370(41,3%)
	$e_{śr}$>E	0,1547	0,1604	0,1487

Zaproponowana metoda opiera się na losowym poszukiwaniu najdłuższej cięciwy. Jako, że wynik końcowy zależny jest od losowania, każde kolejne przebadanie tego samego obrazu daje różne wyniki. Dlatego w trakcie testów, metodę Random przeprowadzono 11 razy. W celu porównania tego testu z pozostałymi wyniki testu podzielono przez 11.

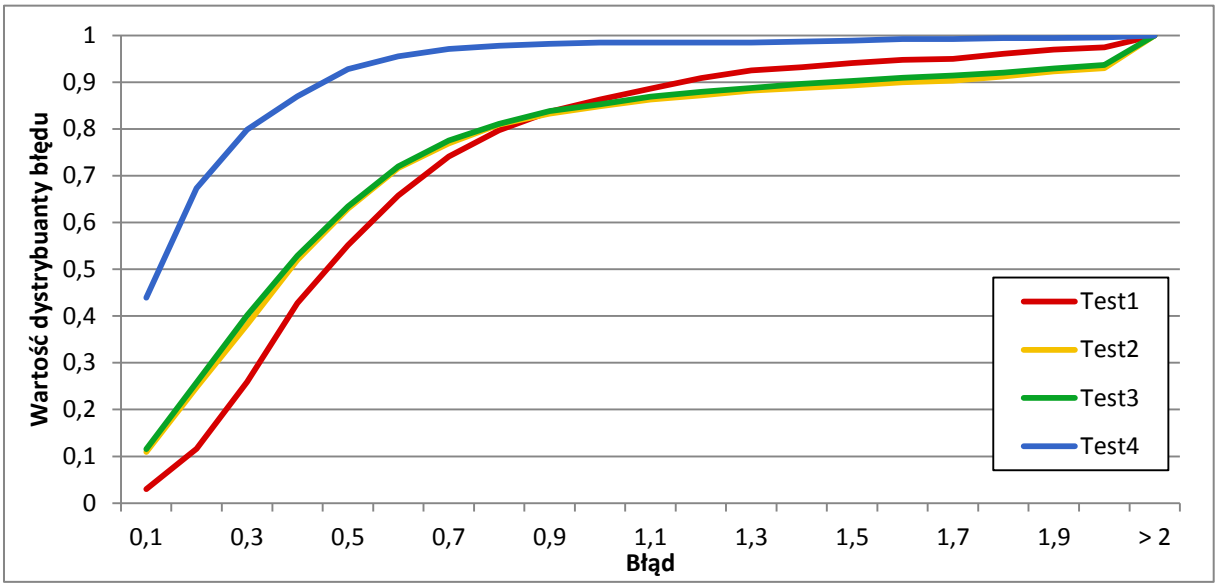
Dodatkowo, w celach dokładniejszego przebadania algorytmu losowego, spośród 11 testów wyłoniono następujące zbiory badawcze:

- Test 1: Dla każdego obrazu wyznaczono średnią błędów dopasowania z 11 przeprowadzonych na nim testów. Na wykresie przedstawiono dystrybuantę średnich wartości błędu;
- Test 2: Wybrano jeden test spośród 11;
- Test 3: Przeprowadzono 11 testów, wykrywając dla każdego obrazu po 11 punktów. Wartości wykresu dystrybuanty podzielono przez 11.
- Test 4: Dla każdego obrazu wybrano najmniejszy błąd spośród przeprowadzonych na nim 11 testów.

Jak można zauważyć na Rys. 44 wynik średni z 11 testów oraz jeden losowo wybrany test prawie są sobie równe. Zaskakujące są jednak pozostałe testy, czyli 1 i 4. Wyniki można zinterpretować w następujący sposób:

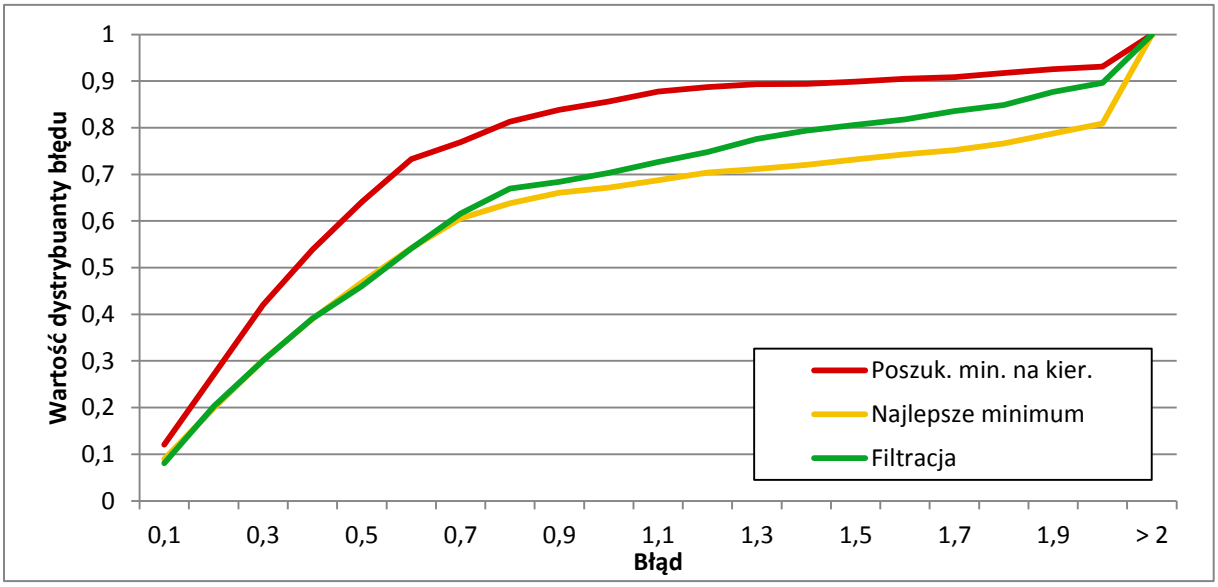
Metoda poszukiwania punktów nadgarstka z użyciem profilu obrazu obróconego, liczy punkty nadgarstka z dużą precyzją, niezależnie od drobnych wahań kąta obrotu. Jeżeli wartości kątów nieznacznie różnią się pomiędzy metodami (Brute Force – Random), otrzymywane wyniki są podobne. W przypadku większych różnic, czyli na przykład w metodzie używającej momentów, która wyznacza czasami całkiem inny kąt, wyniki różnią się znacząco. Jednak można zaobserwować, że w trakcie wyznaczania kąta obrotu metodą losową można natrafić na kąt optymalny, który sprawia, że metoda poszukiwania

nadgarstka, odnajduje pożądane punkty ze znacząco zredukowanym błędem. Niestety, test 4 pokazuje, że podczas analizy danego obrazu z testu można odnaleźć zarówno wynik całkowicie poprawny oraz całkowicie błędny, a średnia wszystkich testów przeprowadzonych na jednym obrazie posiada niską precyzję. Niestety, jako że błąd oparty jest na losowej próbie, parametr odpowiedzialny za podniesienie jakości poszukiwania nadgarstka nie został odnaleziony.



Rys. 44 Dystrybuanta błędu dla różnych interpretacji testów losowych

Podjęto, natomiast, próbę poprawienia metody poszukiwania minimum na profilu. Poniższe wyniki przedstawiają dystrybuantę błędu dla metod opracowanych w ramach poprawy poszukiwania nadgarstka opisanych w rozdziale 4.1.1.



Rys. 45 Dystrybuanta błędu metod poszukiwania lokalnego minimum na profilu w celu wyznaczenia nadgarstka

Wykresy dystrybuant przedstawia Rys. 45. Niestety, zarówno metoda wykorzystująca filtracje profilu aż do momentu uzyskania tylko jednego minimum oraz metoda poszukiwania najlepszego minimum spośród wszystkich dostępnych, nie sprawdziły się. Dla metody

filtracji, powodem jest przede wszystkim przesunięcie punktu minimum towarzyszące wzrostowi rozmiarów filtra. Dla metody poszukiwania najlepszego minimum, trudność w wyznaczeniu odpowiedniego progu decyzyjnego powodowała wybieranie minimów, które oryginalnie były pomijane.

Jako, że niektóre algorytmy poszukiwania punktów charakterystycznych bazują na metodach poszukiwania nadgarstka, do ich implementacji użyto punktów odnajdywanych za pomocą przedstawionego algorytmu bazującego na zliczaniu profilu obrazu odwróconej o kąt najdłuższej cięciwy spośród losowo wybranych par punktów, a następnie wyznaczającego punkt nadgarstka za pomocą metody poszukiwania minimum wzdłuż kierunku z kontrakcją przedziału, oraz z poprawieniem lokalnym wyniku.

5.3.2. Badanie metod wykrywania pozostałych punktów charakterystycznych

Aby w pełni zrozumieć wybór odpowiednich baz obrazów oraz zbiór testów, które przeprowadzono w celu wyznaczenia współczynników jakości wykrycia i lokalizacji punktów charakterystycznych dłoni, należy prześledzić całą ścieżkę prowadzącą do wyłonienia ostatecznego testu. W trakcie serii badań pobocznych wyniknęło wiele wniosków, które składają się na taką, a nie inną strukturę testu ostatecznego i ostatecznych wniosków.

W niniejszym rozdziale testowane będą algorytmy, których opis prezentowany był w poprzednich rozdziałach. Dla przypomnienia oraz w celu wyjaśnienia legendy wykresów, skrócone nazwy użyte w tym rozdziale przedstawia Tab. 5.

Tab. 5 Skrócone nazwy algorytmów używane w trakcie testów

Skrócona nazwa:	Opis:
C2W (ang. contour to distance)	Algorytm opierający się na analizie ekstremów lokalnych funkcji odległości kolejnych punktów konturu do punktu nadgarstka. Opisany w rozdziale 2.3.1 i w [22].
DT (ang. distance transform)	Algorytm opierający się na analizie obrazu transformaty odległościowej. Opisany w rozdziale 2.3.2 i w [79, 81].
TM (ang. template match)	Algorytm dopasowania wzorca koła do maski obrazu. Opisany w rozdziale 2.3.3 i w [78].
SGONG (ang. self-growing and self-organizing neural gas)	Algorytm dopasowujący sieć neuronów do obrazu maski dłoni oraz analizujący sąsiedztwo neuronów. Opisany w rozdziale 2.3.4 i w =[74]=[74].
Autorski	Propozycja autorskiego algorytmu bazującego na obrazie odległościowym. Algorytm opisany w rozdziale 4.

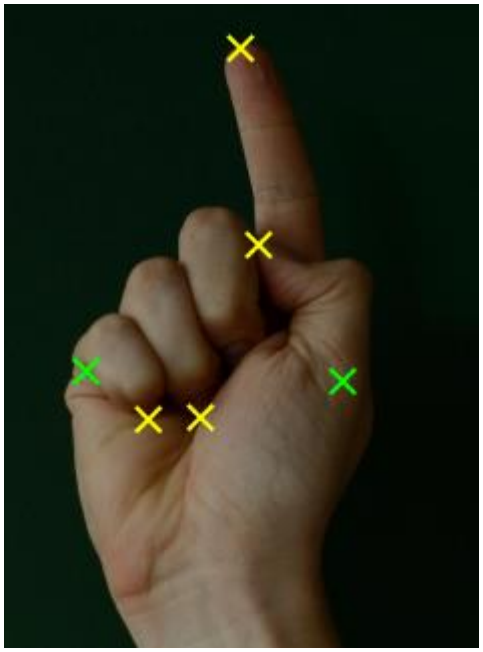
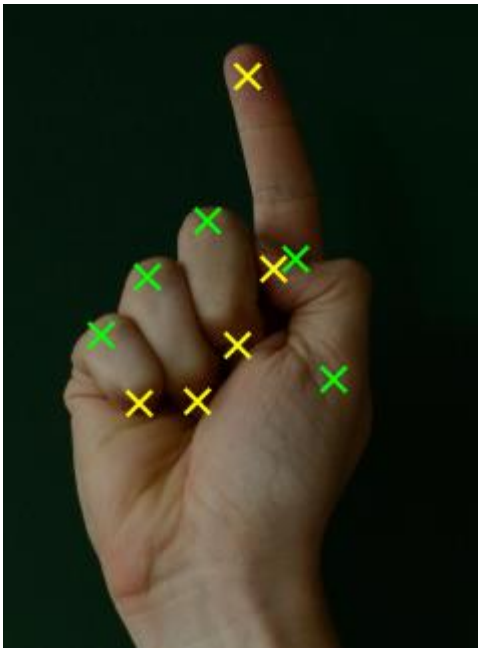
W niektórych testach użyto metody dodatkowej, w celu pokazania dodatkowych wniosków. Metody te na ogół opisane są w odpowiednim miejscu. W szczególności będą to testy błędu ludzkiego (człowiek) lub losowe generowanie punktów w obrazie (ang. random).

♦ **Test bazy hgr2a – błąd ludzki**

Baza hgr2a powstała z pomocą zaledwie 3 osób oraz z użyciem 13 gestów oraz zawiera jedynie 85 obrazów. W przeciwieństwie do bazy hgr1, posiada ona obrazy o wyższej rozdzielczości, a prezentowane gesty są bardziej skomplikowane oraz dobrane tak, aby podkreślać słabe strony algorytmów. Oczywiście, dopiero baza hgr2b stwarza rzeczywiste wyzwanie, jednak hgr2a ma inną dużą zaletę. Mała liczba obrazów pozwoliła na przeprowadzenie testu błędu ludzkiego, który wymagał oznaczenia punktów charakterystycznych tego samego obrazu w różny sposób.

Dodatkowo, mała liczba obrazów jest idealna do wielokrotnych testów, ponieważ testy te przeprowadzane są szybko. Dla przykładu, zakładając średni czas wykonywania równy 0,2 sekundy, przebadanie całej bazy dla 5 algorytmów w przypadku hgr2a to około półtorej minuty, a dla hgr2b to około 10 minut.

Zauważono, że algorytmy, w zależności od wykorzystanej metody, dążą do wyznaczania punktów charakterystycznych w specyficznych miejscach. Dla przykładu, algorytm C2W, analizujący punkty konturu, dążyć będzie do wyznaczania czubków palców na konturze, a algorytm TM, bazujący na dopasowaniu wzorca koła, dążyć będzie do lokalizowania czubka palca w najlepszym punkcie dopasowania, czyli środku opuszka. Tak samo można wyobrazić sobie, że jeden ekspert uzna, że czubek palca znajduje się na końcu paznokcia, a inny, że na środku opuszka. Dlatego, korzystając z najmniejszej bazy, poproszono dwie osoby o skrajnie różne zaznaczanie punktów.

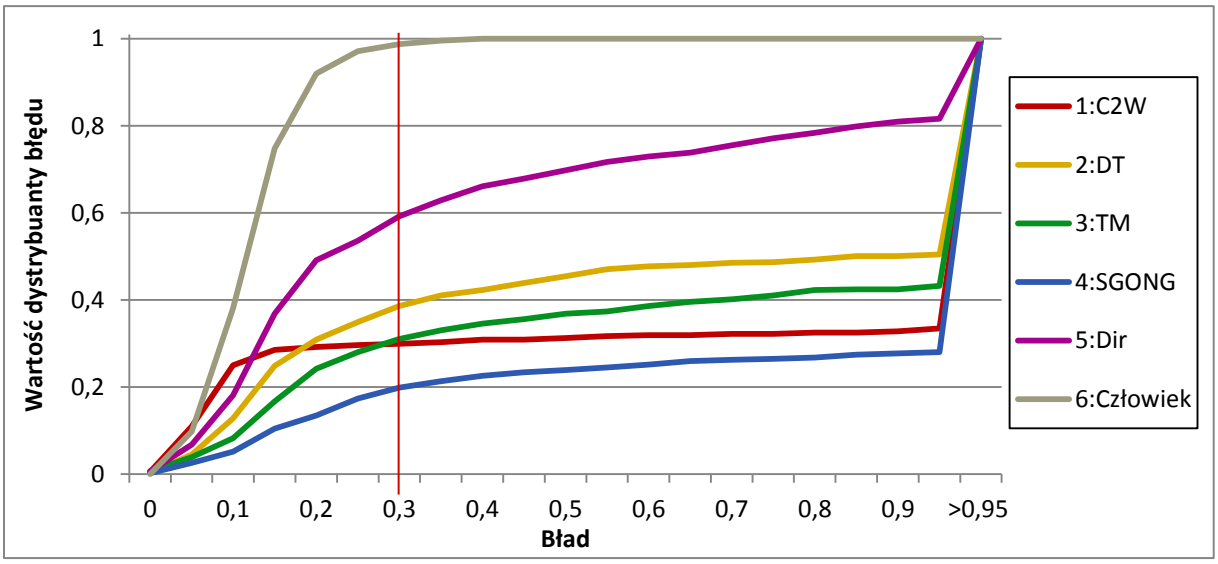
a)  b) 

Rys. 46 Dwa skrajne sposoby oznaczania obrazków przez ekspertów. Na obrazie a) oznaczono tylko punkty widoczne. Nasady palców i czubek jednego palca nie jest widoczny. Na obrazie b) zaznaczono punkty, nawet jeżeli nie były widoczne. Czubki palców oznaczono wewnątrz opuszka.

Punkty we wszystkich bazach oznaczały różne osoby, które nie miały nigdy specjalnych instrukcji odnośnie poprawnego miejsca oznaczenia punktu. Eksperci oznaczali punkty w miejscach, które uznawali za stosowne.

Jednak w celu wyznaczenia błędu ludzkiego, należało stworzyć dwie skrajne różne zasady odnośnie oznaczania punktów charakterystycznych. Zadaniem pierwszej osoby było zaznaczenie punktów charakterystycznych dokładnie na najbardziej wysuniętym punkcie palca, czyli na paznokciu. Dodatkowo, osoba ta proszona była o zaznaczenie punktu tylko w przypadku, jeżeli region był całkowicie widoczny. Druga osoba miała być bardziej tolerancyjna. Zaznaczała ona punkty, nawet jeżeli były częściowo zakryte. Dodatkowo, poproszono ją o zaznaczanie czubków palców wewnątrz opuszka. Przykład skrajnego oznaczania obrazów przedstawiono na Rys. 46.

Celem porównania tych dwóch skrajnych testów było wyznaczenie błędu ludzkiego. Na podstawie tego błędu możliwe jest określenie progu błędu granicznego *E*. Próg *E,* opisany w rozdziale 5.2.2, stanowi granicę między punktami klasyfikowanymi jako poprawnie wykryte TP, oraz błędnie wykrytymi FP. Przeprowadzono zatem test, którego wyniki przedstawia wykres dystrybuanty błędu, zaznaczony na Rys. 47 linią szarą. Wartość funkcji dystrybuanty, dla konkretnego argumentu funkcji, czyli dla błędu, oznacza ile punktów charakterystycznych zostało sklasyfikowanych z błędem mniejszym bądź równym tej wielkości.



Rys. 47 Wykresy dystrybuanty błędów dla testów 5 algorytmów oraz błędu ludzkiego przeprowadzone na bazie hgr2a

W celu ustalenia progu błędu granicznego, do stworzonej metody walidacyjnej, opisanej w rozdziale 5.2.2, zastosowano modyfikację. Potraktowano oznaczenia pierwszego eksperta jako zbiór punktów bazowych, natomiast oznaczenia drugiego eksperta jako zbiór punktów wykrytych. W teście błędu ludzkiego, zbiór GT1 są to dane wykryte, a GT2 to dane wzorcowe. Modyfikacja ta zastosowana jest jedynie do testu błędu ludzkiego. Pozostałe algorytmy testowane są z użyciem niemodyfikowanych danych wzorcowych GT.

Na potrzeby testu błędu ludzkiego, zastosowano modyfikację procedury walidacyjnej, dzięki której porównywane między sobą były konkretne punkty, a nie całe grupy punktów. Oznacza to, że punkty kciuków, porównywane były tylko z punktami kciuków, a nie z całą grupą palców. Przeprowadzenie metody walidacyjnej wyłoniło następujące wyniki, przedstawione w Tab. 6. Pierwszą zauważalną różnicą, jest fakt, że ekspert pierwszy zaznaczył 798 punktów, a ekspert drugi zaznaczył 741. Co więcej, na etapie parowania

punktów, okazało się, że zdania ekspertów były podzielone nie tylko w 57 przypadkach wynikających z różnicy sumy ich wskazań. W sumie 63 punkty nie zostały sparowane na etapie walidacji, co czyni je punktami niewykrytymi, czyli są to punkty FN. Z puli 798 punktów wskazanych, 63 punkty zostały niesklasyfikowane, co daje 735 punktów typu P, które trzeba sklasyfikować za pomocą wartości błędu granicznego *E,* jako poprawne TP lub niepoprawne FP.

Wartość błędu granicznego została dobrana heurystycznie. Po pierwsze, analizując wykres dystrybuanty, wartość błędu granicznego *E* została dobrana we wstępnej fazie nasycenia. Jednocześnie wartość ta została dobrana tak, aby zapewnić precyzje na poziomie około 99%. Wartość błędu granicznego została zatem ustalona jako *E=0,3* oraz naniesiona została na Rys. 47, jako pionowa czerwona kreska.

Jednocześnie warto zauważyć, że precyzje pomiaru można poprawić za pomocą modyfikacji wartości błędu granicznego. Jednak selektywność zależna jest od liczby punktów niewykrytych, co oznacza brak możliwości poprawy wyznaczonego wyniku. Z tej obserwacji wynika wniosek, że oznaczenia punktów charakterystycznych przez inne osoby mogą nie być jednolite, zwłaszcza w przypadku, gdy punkty nie zawsze są widoczne.

Tab. 6 Wyniki procedury walidacyjnej dla ustalenia błędu ludzkiego przeprowadzonego na bazie hgr2a.

Algorytm	GT1	GT2	Punkty bez pary	Sparowane punkty		Miary statystyczne		
				TP	FP	TPR	PPV	F1
Człowiek	798	741	63	726	9	92,02%	98,77%	95,28%

Najlepszym rozwiązaniem byłoby oznaczenie każdego obrazu przez wielu ekspertów, a następnie, w trakcie testowania algorytmu, porównywać wykryte punkty z najbliższym punktem wskazanym przez jednego z ekspertów. Niestety, oznaczanie punktów charakterystycznych jest zajęciem czasochłonnym, dlatego większość obrazów została oznakowana tylko raz. Wyznaczenie skrajnych różnic w oznaczaniu punktów charakterystycznych przez dwóch ekspertów, posłużyło do wyznaczenia wartości referencyjnej błędu. Dystrybuanta przedstawiona na wykresie 6 z Rys. 47 odwzorowuje błąd ludzki.

Dodatkowo na Rys. 47 oraz w Tab. 7 zaprezentowano wyniki wykrywania punktów charakterystycznych dla 5 opisanych algorytmów.

Tab. 7 Wyniki procedury walidacyjnej algorytmów przeprowadzone na bazie hgr2a.

Algorytm	GT	Wy-kryte	Punkty bez pary	Sparowane punkty		Miary statystyczne		
				TP	FP	TPR	PPV	F1
C2W	741	250	492	216	33	30,51%	86,75%	45,14%
DT	741	386	355	282	103	44,27%	73,25%	55,19%
TM	741	381	379	223	139	37,04%	61,60%	46,27%
SGONG	741	218	524	144	73	21,56%	66,36%	32,54%
Autorski	741	722	92	430	219	82,38%	66,26%	73,44%

Gdy przebieg wykresu szybko stabilizuje się, oznacza to, że wykryte przez algorytm punkty posiadają małą wartość błędu. Co więcej, jeżeli wartość dystrybuanty w *e = E* nie jest

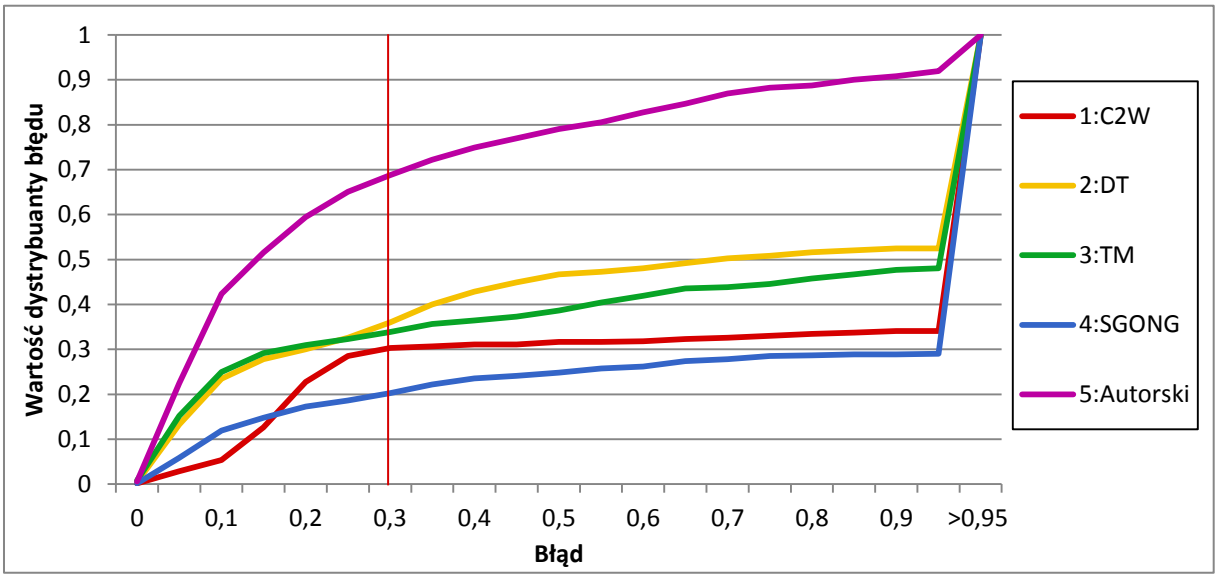
znacząco niższa niż w $e \rightarrow \infty$, czyli $F(E) \approx \lim_{e\to\infty} F(e)$, wtedy precyzja będzie zbliżona do wartości *100%*. Przykładem może być algorytm C2W, który wykrywa mało punktów, jednak punkty sparowane rzadko posiadają błąd dopasowania większy niż błąd graniczny.

Inną cechą dobrego wykresu dystrybuanty jest osiągnięcie jak najwyższej wartości dla punktu błędu granicznego. Jako, że wykres przedstawia rozkład liczby punktów sklasyfikowanych z konkretnym błędem, oznacza to, że im więcej punktów zostanie sklasyfikowanych poprawnie, tym mniej punktów zostanie niesklasyfikowanych lub błędnie sklasyfikowanych, co oznacza wyższą selektywność testu. Selektywność zatem zależy od wartości dystrybuanty w *e=E* oraz liczby punktów wzorcowych *GT*. Jeżeli $F(E) = GT$, to liczba punktów niewykrytych będzie równa 0, a selektywność równa będzie 100%. Algorytmem o najwyższej selektywności jest algorytm autorski, wykorzystujący obraz kierunkowy.

Najlepszy algorytm zatem będzie cechował się wysoką precyzją oraz wysoką selektywnością. Miarę testu przedstawia wynik F1, czyli miara średniej harmonicznej precyzji i selektywności. Jak widać z Tab. 6, proponowany algorytm, cechuje się najwyższą selektywnością oraz wysoką precyzją . Miara testu F1 jest zatem najwyższa dla algorytmu autorskiego.

♦ Test bazy hgr2a – analiza algorytmów

Rys. 47, przedstawia wyniki algorytmów, przyrównane na etapie walidacji do wskazań jednego eksperta. Jako, że dla tej bazy stworzono dwa zbiory wzorcowe, przeprowadzono walidację wyników przyrównując je do wskazań drugiego eksperta. Wyniki przedstawione są na Rys. 48.



Rys. 48 Wykresy dystrybuanty dla wyników 5 algorytmów działających na bazie hgr2a, przyrównanych do wskazań innego eksperta

Porównując wizualne aspekty wykresów widać, że po przekroczeniu błędu granicznego, wykresy dystrybuanty są prawie identyczne dla wskazań obu ekspertów. Znaczące różnice

występują po lewej stronie, czyli w granicach błędu akceptowalnego. Jest to zatem dodatkowe wykazanie tego, że wartość błędu granicznego została dobrana poprawnie.

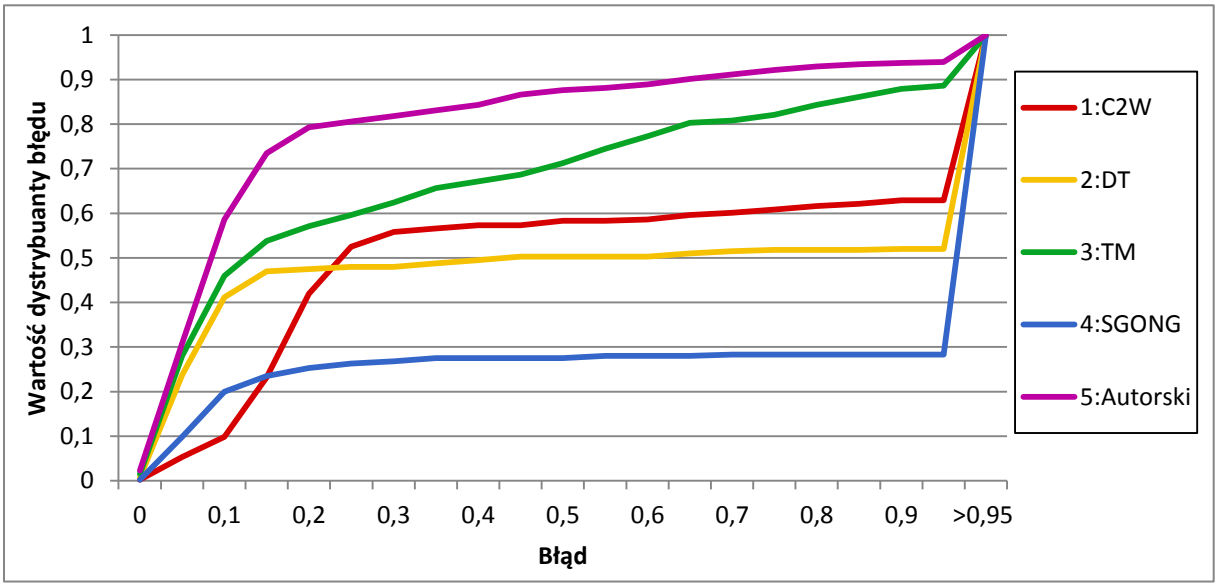
Różnice te widoczne są zwłaszcza dla algorytmu C2W, ponieważ algorytm ten wykrywa punkty na konturze, a pozostałe algorytmy, wykrywają je po środku opuszka. Widać zatem, że w przypadku wskazywania punktów na czubkach palców, algorytm 1 ma dużą precyzję dla niskich wartości błędu, a w przypadku wskazywania punktów na środku opuszka, algorytm 3 ma dużą precyzję dla niskich wartości błędu.

Dodatkowo widać też, że w regionie błędu akceptowalnego, dla skrajnych wskazań ekspertów, wykres 1 przybiera postać wykresu 3, a wykres 3 postać wykresu 1. Oznacza to, że jeżeli błąd graniczny dobrany zostałby z niższą wartością, wtedy dokładność algorytmu zależałaby od sposobu oznaczania punktów przez eksperta, a nie od ich faktycznej lokalizacji.

- **Test bazy hgr2a – Porównanie punktów palców**

Analizując wyniki poprzedniego testu, niepokojące może wydawać się, że miary statystyczne prezentowane w Tab. 6 Tab. 7 są niskie. Większość algorytmów nie uzyskuje nawet wyniku powyżej *50%* poprawnie wykrytych punktów. Powodem może być fakt, że walidacja przeprowadzana jest dla dwóch grup punktów, czyli zbioru punktów czubków palców oraz zbioru punktów nasady palców. Niestety, część algorytmów nie wykrywa nasady palców. Są to algorytmy C2w i TM. W przypadku, gdyby algorytmy te wykryły wszystkie czubki palca idealnie, zapewniając precyzję na poziomie *100%,* ich selektywność nie przekroczyłaby *50%,* dając wynik miary testu *F1=75%.* Jest to założenie dla teoretycznej bazy obrazów, w której wszystkie możliwe punkty były widoczne i zostały oznaczone.

Dlatego, w celu porównania wyników wykrycia samych czubków palców, przeprowadzono test, w którym punkty nasady palców nie poddawane były walidacji. Wynik przedstawione są w Tab. 8 i na Rys. 49.



Rys. 49 Wykresy dystrybuanty dla walidacji jedynie punktów czubków palców, testowanych na bazie hgr2a

Analizując wyniki widać, że faktycznie dla algorytmów C2W i TM, zaobserwowano poprawę selektywności oraz większe wartości funkcji dystrybuanty. Jednak ogólne wnioski pozostają takie same. Oznacza to, że nadal proponowany algorytm wykrywa najwięcej punktów charakterystycznych, co potwierdza wysoka selektywność, jednocześnie posiadając wystarczającą precyzję aby uzyskać najwyższy wynik testu.

Tab. 8 Wyniki walidacji jedynie punktów czubków palców, testowanych na bazie hgr2a

Algorytm	GT	Wy-kryte	Punkty bez pary	Sparowane punkty		Miary statystyczne		
				TP	FP	TPR	PPV	F1
C2W	414	258	156	220	38	53,14%	85,27%	65,48%
DT	414	209	205	187	22	45,17%	89,47%	60,03%
TM	414	396	18	241	146	58,21%	62,27%	60,17%
SGONG	414	115	299	105	10	25,36%	91,30%	39,70%
Autorski	414	398	27	315	72	92,11%	81,40%	86,42%

Dodatkowo, można zaobserwować, że algorytm 3 ma najbardziej zbliżony wykres dystrybuanty do algorytmu 5. Powodem jest fakt, że algorytm 5 został opracowany w oparciu o metodę dopasowania wzorca. Rozszerzenie algorytmu 5 sprawiło, że dodatkowe punkty wewnątrz maski zostały wykryte, powodując wysokie wartości dystrybuanty dla małych błędów, jednocześnie fałszywie wykryte punkty zostały odrzucone, sprawiając, że suma fałszywych punktów jest niższa.

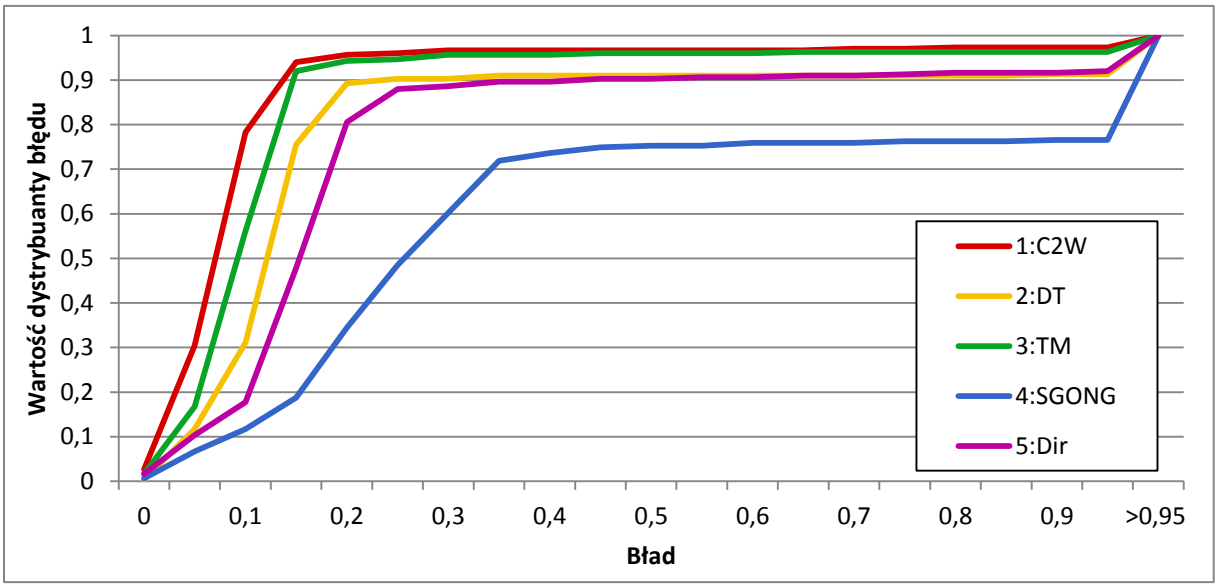
Nadal jednak powstaje pytanie, dlaczego średnio co trzeci punkt nie zostaje wykryty lub poprawnie sklasyfikowany? Aby odpowiedzieć na to pytanie przeprowadzono test na wybranych obrazach, których cechy charakterystyczne są dobrze wyeksponowane.

- **Test przeprowadzony na wybranych obrazach**

Dotychczasowe testy mogą prowadzić do wniosków, że żaden z prezentowanych algorytmów nie jest wystarczająco skuteczny. Dodatkowo mogą być sprzeczne z wynikami prezentowanymi w publikacjach. Niestety, większość z algorytmów ma nakładane ograniczenia odnośnie sposobu prezentowania dłoni. Najwięcej ograniczeń posiada algorytm SGONG, który zakłada pionową orientację dłoni, oraz brak widocznego łokcia na obrazie. Jednak na etapie tworzenia bazy stawiano nacisk, aby obrazy były jak najbardziej różnorodne, oraz aby przedstawiały gesty skomplikowane. Gesty pokazywane są w orientacji pionowej, poziomej, przodem, tyłem, od lewej oraz od prawej. Nie istnieją ograniczenia odnośnie długości rękawa, tak więc niektóre maski regionu skóry przedstawiają całą dłoń aż do ramienia, a niektóre jedynie do nadgarstka. W przypadku, gdy testowany algorytm wymaga pewnych założeń, które nie są spełniane przez obrazy wejściowe, jego wyniki są błędne.

W celu przedstawienia, że każdy z algorytmów przy odpowiednio dobranych warunkach jest w stanie rozpoznać dłoń z dużą dokładnością, spośród wszystkich obrazów wybrano 65 obrazów dłoni, które są łatwe do rozpoznania. Są to głównie obrazy przedstawiające wszystkie palce, szeroko rozpostarte. Wybrane gesty najczęściej odnoszą się do gestu

oznaczającego „4” lub „5” w języku migowym. Wyniki testów przedstawione są na Rys. 50 oraz w Tab. 9.



Rys. 50 Wykresy dystrybuanty wyników testu łatwego

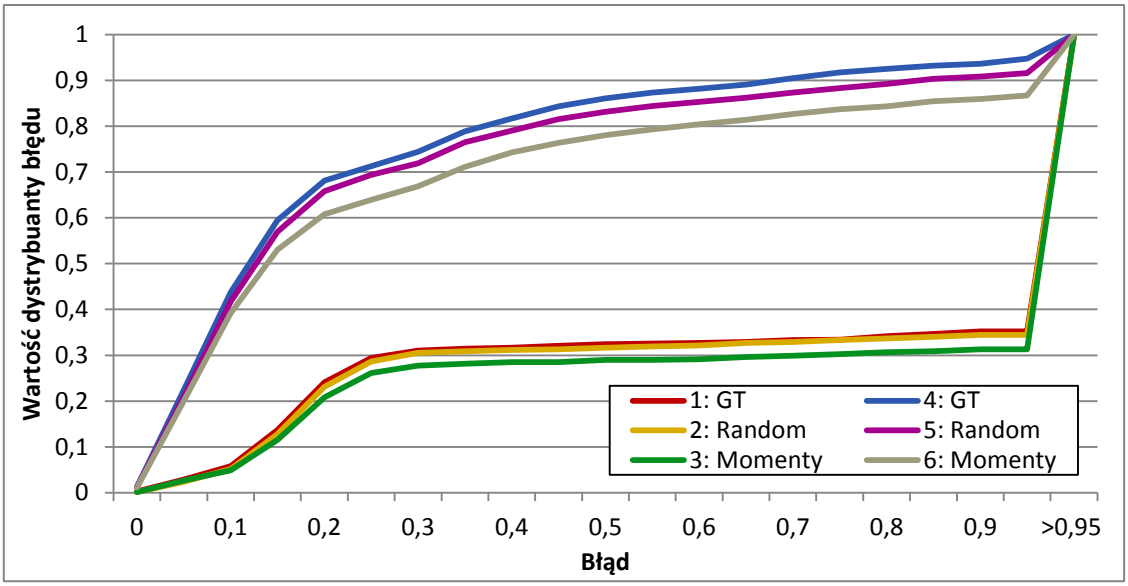
Dla tak dobranego zestawu obrazów, każdy algorytm cechuje się wysoką skutecznością. Niestety, dla algorytmu SGONG, 11 z 64 obrazów nie spełniało założeń odnośnie orientacji dłoni, przez co jego wynik jest najsłabszy. Dodatkowo algorytmy TM i SGONG wykryły zbyt dużo punktów nadmiarowych, dla których nie wykryto pary z punktami bazowymi, co w znaczący sposób obniżyło skuteczność. Jednak prezentowane wyniki testu pokazują, że każdy algorytm, przy odpowiednio dobranym zestawie obrazów jest w stanie wykrywać punkty charakterystyczne dłoni z wysoką skutecznością.

Tab. 9 Wyniki walidacji testu łatwego

Algorytm	GT	Wykryte	Punkty bez pary	Sparowane punkty		Miary statystyczne		
				TP	FP	TPR	PPV	F1
C2W	295	291	6	281	10	97,91%	96,56%	97,23%
DT	295	278	17	271	7	94,10%	97,48%	95,76%
TM	295	310	31	281	13	90,06%	95,58%	92,74%
SGONG	295	251	50	213	32	80,99%	86,94%	83,86%
Autorski	295	309	37	265	21	87,75%	92,66%	90,14%

♦ Test wpływu lokalizacji nadgarstka

W poniższym teście przebadano wpływ metody poszukiwania nadgarstka na jakość działania algorytmów poszukiwania pozostałych punktów charakterystycznych. Test przeprowadzono dla dwóch algorytmów, których wyniki zależą od lokalizacji nadgarstka, czyli dla C2W oraz algorytmu autorskiego. Dodatkowo przeprowadzono drugą serię testów, w których punkty nadgarstka nie były wykrywane, lecz były pobrane z bazy punktów wzorcowych. Wyniki przedstawione są na Rys. 51 oraz w Tab. 10.



Rys. 51 Wykresy dystrybuanty wyników testu wpływu lokalizacji nadgarstka. Testy 1-3 przedstawiają wyniki algorytmu C2W. Testy 4-6 przedstawiają wyniki algorytmu autorskiego. Punkty nadgarstka wyznaczano metodami losowania najdłuższej przekątnej (random), wyznaczając momenty oraz pobierając punkty z bazy GT.

Zaobserwować można, że algorytm wykrywania nadgarstka ma nieznaczny wpływ na jakość wykrywania pozostałych punktów charakterystycznych. Obniżenie miar statystycznych metod wyznaczania punktów nadgarstka względem pobierania punktów wzorcowych wywodzi się z kilku przypadków, w których nadgarstek nie został odnaleziony. W przypadku, gdy nadgarstek został odnaleziony, jego błąd dopasowania nie wpłynął na wynik błędu dopasowania punktów charakterystycznych.

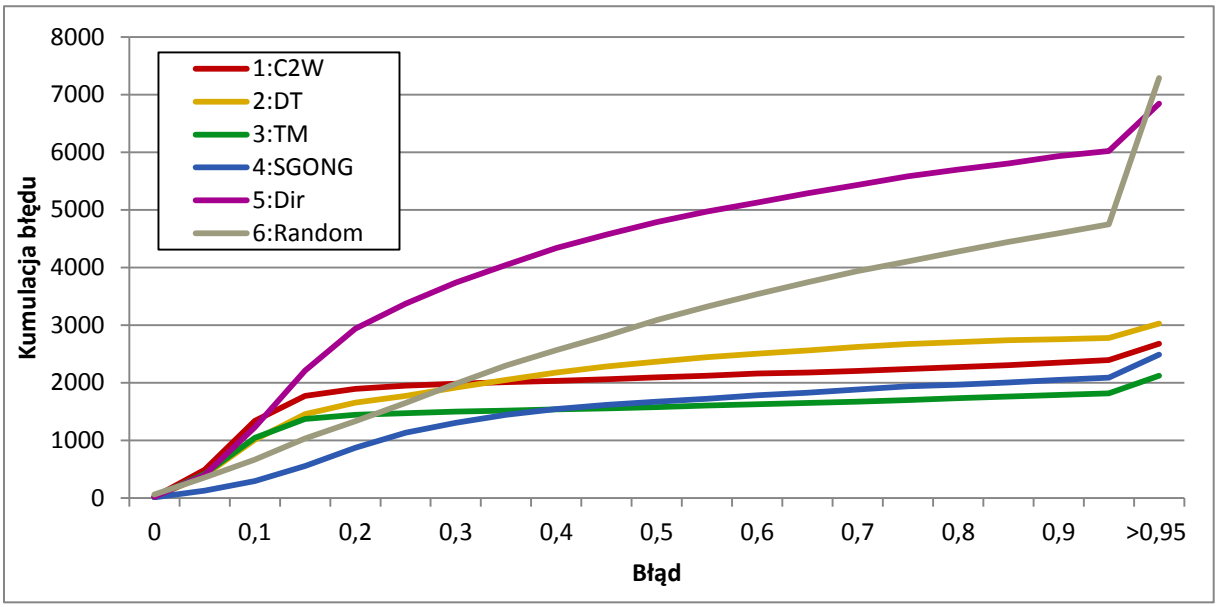
Tab. 10 Wyniki walidacji wpływu lokalizacji nadgarstka

Algorytm	GT	Wy-kryte	Punkty bez pary	Sparowane punkty		Miary statystyczne		
				TP	FP	TPR	PPV	F1
C2W:								
GT	826	288	538	245	43	31,29%	85,07%	45,75%
Random	826	286	540	242	44	30,95%	84,62%	45,32%
Momenty	826	258	568	220	38	27,92%	85,27%	42,07%
Autorski:								
GT	826	810	40	582	204	93,57%	74,05%	82,67%
Random	826	796	54	562	210	91,23%	72,80%	80,98%
Momenty	826	760	90	736	270	83,81%	63,32%	72,14%

Różnice w wynikach lokalizacji punktów charakterystycznych dłoni dla różnych metod poszukiwania nadgarstka nie są duże. W dalszych testach użyto autorskiej metody wyznaczania nadgarstka z użyciem przekątnej konturu ustalonej za pomocą testu losowego nazwanej random. Ponadto zaobserwowano, że parametry sterujące algorytmem wyznaczania nadgarstka nie mają znaczącego wpływu na wyniki tego testu.

♦ **Test bazy hgr1 oraz test losowy**

Baza hgr1 jest pierwszą bazą, która powstała jeszcze do celów porównywania algorytmów detekcji i segmentacji skóry. Baza ta posiada najwięcej gestów i pokrywa cały polski alfabet palcowy. Niestety, obrazy są niskiej rozdzielczości co w niektórych przypadkach powoduje zbyt niską precyzję. Wyniki przeprowadzonych na tej bazie testów przedstawia Rys. 52, który prezentuje wykres kumulacji błędów, oraz Rys. 53, który prezentuje normalizacje wykresów z Rys. 52, gdzie wartość 1 odpowiada liczbie wykrytych punktów dla danego algorytmu. Wyniki walidacji przedstawia Tab. 11.



Rys. 52 Wykres kumulacji błędu w testach na bazie hgr1

Dodatkowo, w trakcie testów zaproponowano wprowadzenie algorytmu wybierającego losowe punkty z wnętrza maski dłoni, w celu ustalenia najgorszego przypadku detekcji. Wyniki algorytmu losowego przedstawia algorytm Random. Wyniki pokazują najgorszy przypadek, czyli pełne losowanie wszystkich punktów. Dla każdego obrazu dobrano 5 losowych punktów czubków palców oraz 5 losowych punktów nasady. Zatem dla 897 zdjęć z bazy, wybrano po 10 losowych punktów.

Tab. 11 Wyniki walidacji testu na bazie hgr1

Algorytm	GT	Wy-kryte	Punkty bez pary	Sparowane punkty		Miary statystyczne		
				TP	FP	TPR	PPV	F1
C2W	7222	2927	4295	1964	936	31,38%	67,72%	42,89%
DT	7222	3043	4179	1903	1113	31,29%	63,10%	41,83%
TM	7222	2547	4675	1479	623	24,03%	70,36%	35,83%
SGONG	7222	2544	4678	1294	1185	21,67%	52,20%	30,62%
Autorski	7222	8292	1070	3715	3104	77,64%	54,48%	64,03%
Random	7222	8970	1748	1001	6221	36,41%	13,86%	20,08%

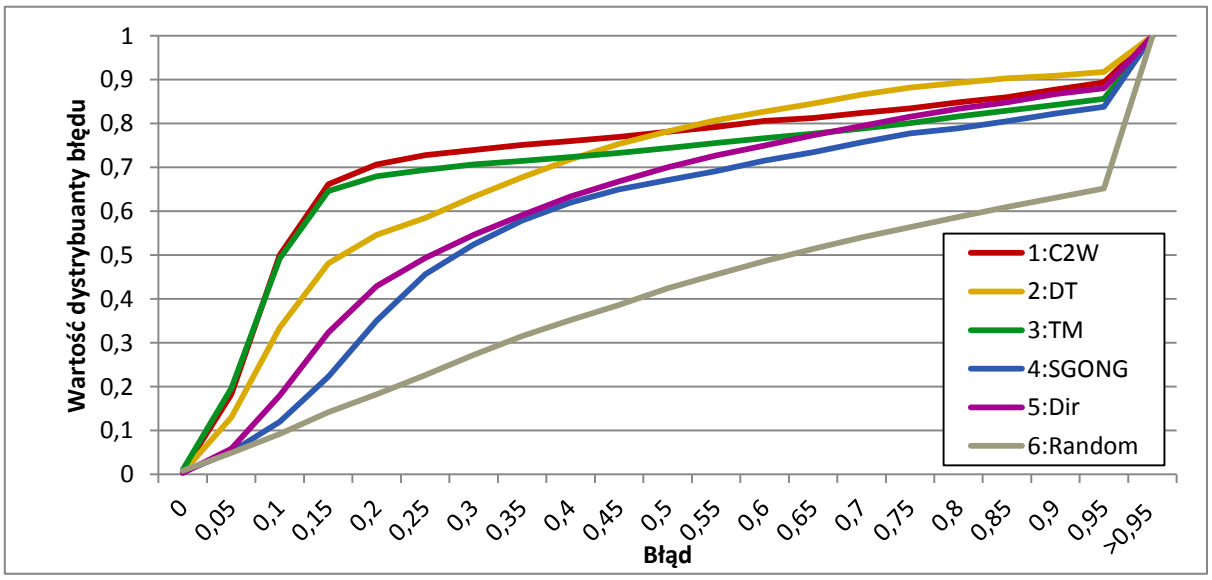
W bazie oznaczonych było 7222 punktów, co powoduje, że wszystkie 7222 punkty zostały sparowane. Tylko 1001 z nich zakwalifikowało się do kryterium błędu granicznego *e<E=0,3.* Z

drugiej strony, punkty nie wskazane przez ekspertów, w tym przypadku, stają się punktami nadmiarowymi, czyli niesparowanymi. Tych punktów było 1748. W końcowym wyniku widać niską precyzję algorytmu losowego, oraz selektywność na wysokim poziomie. Niska precyzja jest zrozumiała, ponieważ losowo dobierane punkty rzadko są uznawane jako poprawne. Wysoka selektywność, na poziomie wyższym niż selektywność czterech wybranych z literatury rozwiązań, spowodowana jest tym, że algorytm losowy skonstruowany został z założeniem, że szukane jest 5 palców. Aby w pełni odwzorować algorytm losowy, należałoby losować liczbę wykrywanych palców, co spowodowałoby znaczny spadek liczby sparowanych punktów, a co za tym idzie spadek selektywności.

Z testów na dużej bazie wynikają podobne wnioski jak z testów na bazie mniejszej. Widać, że proponowany algorytm nadal cechuje się dużą selektywnością, jednak jego precyzja w znaczący sposób spadła. Spadła również precyzja pozostałych algorytmów, głównie z powodu małego rozmiaru zdjęć.

W celu lepszego zobrazowania granicznej precyzji wynikającej z testu losowego, wykresy dystrybuanty znormalizowano tak, że maksymalną wartością dla każdego wykresu jest wartość 1 odpowiadająca liczbie wykrytych punktów dla danego algorytmu. Aby uzyskać taki wykres, każdą wartość dystrybuanty *F(n)* podzielono przez wartość ostatnią *F(N).*

Znormalizowany wykres odwzorowuje precyzję. Na wykresie tym widać przede wszystkim, że algorytm losowy przedstawia się jako linia prosta, o stałym nachyleniu. Oznacza to, że około tyle samo punktów zostało wykrytych w każdym przedziale błędów, a rozkład kumulacji błędu jest jednolity.

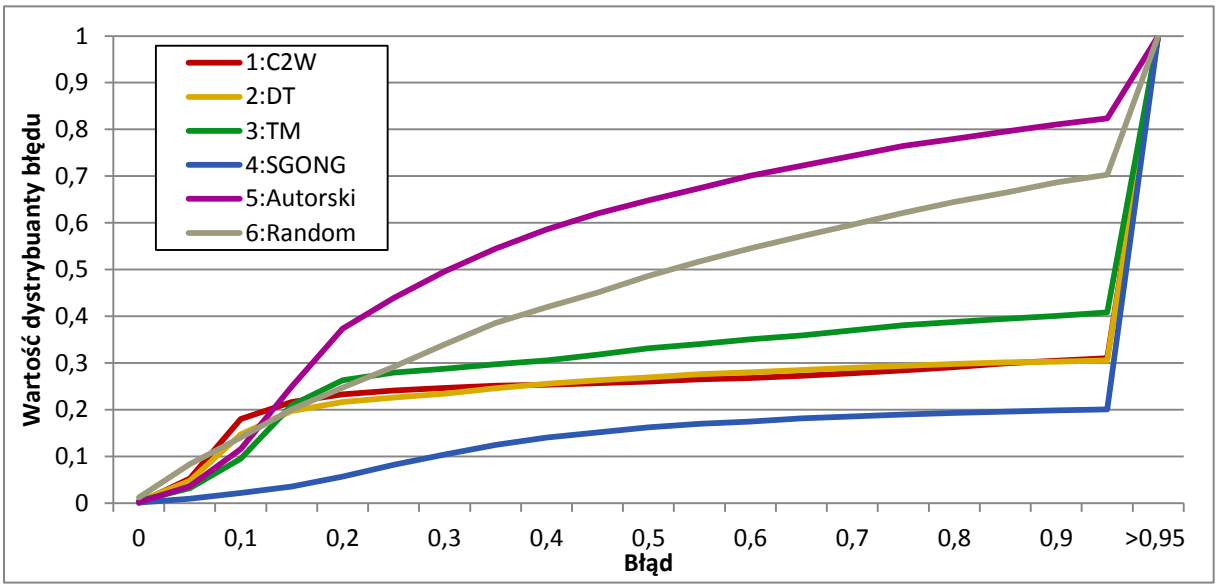


Rys. 53 Wykres dystrybuanty wyników testu na bazie hgr1

Rys. 47 przedstawiający wykres dystrybuanty błędu ludzkiego (wykres nr 6) oraz Rys. 53 przedstawiający wykres dystrybuanty algorytmu losowego (wykres nr 6), stanowią odpowiednio granicę testu idealnego oraz najgorszego. W interpretacji graficznej oznacza to, że wykresy dystrybuanty wyników testów znajdować się będą pomiędzy wykresami granicznymi oraz nie przewiduje się aby wykres dystrybuanty wyników testu przecinał którykolwiek z tych dwóch wykresów granicznych.

♦ **Test bazy hgr2b**

Ostatecznym testem był test przeprowadzony na bazie hgr2b. Jest to baza specjalnie dobrana w taki sposób, aby stworzyć swoiste wyzwanie dla algorytmów poszukujących punkty charakterystyczne. Obrazy wykonane są w dużej rozdzielczości, więc posiadają dużo szczegółów, jednak prezentowane gesty bywają do siebie bardzo podobne. Czasem tak podobne, że maski dłoni dwóch różnych gestów są prawie identyczne.



Rys. 54 Wykresy dystrybuanty błędu testów na bazie hgr2b

Wyniki testu na bazie hgr2b przedstawia Rys. 54 oraz Tab. 12. Baza ta zawiera zestaw gesty skomplikowane, trudne do rozróżnienia. Dlatego wyniki tego testu są najgorsze spośród wszystkich prezentowanych wcześniej. Jednak poza spadkiem głównie selektywności, wzajemna relacja między algorytmami pozostaje podobna.

Zarówno te, jak i poprzednie testy wykazały podobne wyniki. Precyzja algorytmu proponowanego jest porównywalna z precyzją algorytmów zaczerpniętych z literatury. Dużą przewagą jest natomiast selektywność, ponieważ algorytm działający na podstawie obrazu kierunkowego analizuje wnętrze maski, dzięki czemu wykrywa znacznie więcej punktów niż pozostałe algorytmy.

Tab. 12 Wyniki błędu testów na bazie hgr2a

Algorytm	GT	Wy-kryte	Punkty bez pary	Sparowane punkty		Miary statystyczne		
				TP	FP	TPR	PPV	F1
C2W	4511	1647	2921	1124	508	27,79%	68,87%	39,60%
DT	4511	1891	2635	1053	823	28,55%	56,13%	37,85%
TM	4511	2660	2211	1295	1005	36,94%	56,30%	44,61%
SGONG	4511	1026	3492	472	547	11,91%	46,32%	18,94%
Autorski	4511	5272	1860	2256	1917	54,81%	54,06%	54,43%

Niestety widać jednak, że algorytm 5 cechuje się wykrywaniem zbyt dużej liczby punktów. Jego główną wadą jest wykrywanie punktów nadmiarowych. Selektywność zależna jest od sumy punktów niesparowanych, czyli takich, które wykryte zostały nadmiarowo. Niska

wartość selektywności większości algorytmów wynika ze zbyt małej liczby wykrytych punktów. Zbyt dużo wykrytych punktów również powoduje nie dobieranie ich w pary, co skutkuje spadkiem selektywności w przypadku proponowanego algorytmu 5.

5.3.3. Wnioski

Po przeanalizowaniu pełnej ścieżki badań, dla każdego z opisywanych i testowanych algorytmów, sformułowano następujące wnioski.

Algorytm C2W, opierający się na analizie ekstremów lokalnych funkcji odległości kolejnych punktów konturu do punktu nadgarstka, cechuje się wysoką precyzją. Większość wykrytych punktów lokalizowana jest w poprawnych miejscach, powodując, że fałszywych pozytywów jest mało. Algorytm ten, poprzez swoją strukturę, nadaje się do odnajdywania samodzielnych wystawionych palców. Niestety nie wykrywa on nasad palców ani zgiętych palców. W przypadku, gdy palce są połączone, algorytm traktuje taki region jako całość, wykrywając tylko jeden punkt. To powoduje, że mimo wysokiej precyzji, algorytm ma małą selektywność, przez co ogólna ocena algorytmu nie jest zbyt wysoka. W przypadku porównywania tylko czubków palców, algorytm ten zyskuje na selektywności, jednak nie na tyle by stać się najlepszym algorytmem. Procedura odnajdywania punktów zależna jest od lokalizacji nadgarstka, dlatego w przypadku błędnego odnalezienia lub orientacji nadgarstka, odnalezione punkty charakterystyczne mogą być oznaczane na przykład na łokciu.

Algorytm DT, opierający się na analizie obrazu transformaty odległościowej nie wyróżnia się szczególnymi wynikami. Jego ocena znajduje się pomiędzy innymi metodami. Główną zaletą algorytmu jest identyfikacja palców jako całości oraz wyznaczanie nasady i czubka palca. Dzięki temu dodatkowe punkty nasady są wykrywane, jednak algorytm analizuje tylko palce wyodrębnione. W przypadku zwartej dłoni, algorytm 1 wykryłby taką strukturę jako jeden palec, jednak algorytm 2 nie wykryje nic. Zaletą algorytmu jest wyznaczanie punktów charakterystycznych niezależnie od nadgarstka. Wadą natomiast jest fakt, że region dłoni wyznaczany jest za pomocą maksimum z transformaty odległościowej, która może być pomylona z regionem śródręcza.

Algorytm TM, dopasowujący wzorzec koła do maski obrazu, jest algorytmem, który posiada wysoką selektywność, mimo faktu, że nie wykrywa on punktów nasady palców. W przypadku, gdy palce są połączone, dla dostatecznie dużej rozdzielczości obrazu istnieje szansa na zlokalizowanie czubków każdego palca. Niestety, ciężko jest ustalić granicę odrzutu fałszywych wykryć, dlatego algorytm czasem odnajduje czubki palców na wystających elementach maski, takich jak, na przykład, łokieć lub ostre zakończenie rękawa. Koleją wadą jest fakt, że w przypadku zgiętych palców, algorytm wykrywa czubek palca w miejscu widocznej wypukłości kostki zdjętego palca. Algorytm ten jest niezależny od regionu nadgarstka, co akurat jest jego poważną wadą, ponieważ analizuje maskę dłoni jako całość. Nie posiada żadnych kryteriów filtracji regionu przedramienia, który często widoczny jest na obrazach z bazy gestów.

Algorytm SGONG, dopasowujący sieć neuronów do obrazu maski dłoni oraz analizujący sąsiedztwo neuronów, zapowiadał się na etapie implementacji bardzo obiecująco. Główną

zaletą algorytmu jest identyfikowanie palców jako całości poprzez wyznaczanie nasady oraz czubka palca. Niestety algorytm ten jest skomplikowany obliczeniowo, a każda próba optymalizacji obniża precyzję ostatecznego wyniku. Samo zaimplementowanie algorytmu oraz odpowiednie dobranie parametrów było swoistym wyzwaniem. Parametry zostały dobrane w sposób eksperymentalny, a rezultatem jest nierównomierne rozprzestrzenianie się sieci neuronów, co negatywnie wpływa na precyzję. Nie jest to jednak główna wada algorytmu, ponieważ wykres dystrybuanty mimo, że stabilizuje się długo, to nadal stabilizuje się przed wartością błędu granicznego. Główną wadą algorytmu, oraz głównym powodem, dla którego prawie w każdym teście algorytm ten wypadł najgorzej, jest sama jego struktura. Przede wszystkim algorytm, poprzez szereg założeń, nie jest w stanie poprawnie określić regionu śródręcza, jeżeli dłoń nie jest ustawiona pionowo. Dodatkowo algorytm zależny jest od rozdzielczości obrazu, co powoduje, że dla obrazów zawierających region śródręcza, palce mogą być zbyt małe, dlatego mogą zostać niezidentyfikowane. Ostatecznie, algorytm ten wykrywa często palce na odosobnionych regionach maski, takich jak brzegi rękawów lub łokieć. Niestety, w prezentowanych testach algorytm ten nie posiada ani wysokiej precyzji, ani wysokiej selektywności, ani szybkiego czasu obliczeniowego.

Algorytm autorski, który jest propozycją rozwiązania bazującego na obrazie odległościowym, w większości testów wykazywał najlepsze wyniki, cechując się wysoką selektywnością, precyzją i miarą testu F1. Jego główną zaletą jest analiza punktów wewnątrz maski, co powoduje odnajdywanie punktów palców zgiętych. Algorytm dostarcza pełną informację odnośnie czubka oraz nasady palca. Posiada on również kryterium filtracji punktów nie znajdujących się w regionie dłoni. Niestety, brakuje poprawnie zdefiniowanego kryterium filtracji punktów błędnie wykrytych, co powoduje wykrywanie zbyt dużej liczby punktów nadmiarowych. Wpływa to negatywnie na selektywność. Precyzja algorytmu nie jest najwyższa, ponieważ dodatkowe punkty wewnątrz maski często odnajdywane są ze zbyt dużym błędem. Jednak przez to, że algorytm wykrywa najwięcej poprawnych punktów charakterystycznych, cechując się wysoką selektywnością oraz porównywalną z innymi algorytmami precyzją, sprawia że jest to algorytm o najlepszym wyniku dla każdego przeprowadzonego testu, który określa miara testu F1. Potwierdza to tezę postawioną w rozdziale 1.6, że wykorzystane w pracy metody sztucznej inteligencji pozwalają na dokładniejszą lokalizację i detekcję wybranych punktów charakterystycznych dłoni.

Dodatkowym potwierdzeniem precyzji proponowanego algorytmu, są badania części odpowiedzialnej za detekcję nadgarstka. W prezentowanych algorytmach nadgarstek był albo pomijany, co powodowało błędną detekcje regionu śródręcza, albo wykrywany z niską precyzją. Badania potwierdzają konieczność poprawnej identyfikacji regionu dłoni, w celu poprawnej lokalizacji punktów charakterystycznych oraz w celu filtracji wykrytych punktów poza obszarem dłoni. Zaproponowana metoda poszukiwania punktów nadgarstka, stanowi trzon najlepszych algorytmów, czyli proponowanego algorytmu 5, oraz została wprowadzona jako modyfikacja algorytmu 1.

W trakcie prezentacji wstępnych wyników, najbardziej dyskusyjnym tematem była procedura walidacyjna. Celem algorytmu jest poprawne odwzorowanie ułożenia dłoni oraz

lokalizacja palców, dlatego algorytmy nie identyfikujące nasady palców powinny być odpowiednio karane. Mimo to przeprowadzony został test pokazujący wyniki walidacji jedynie czubków palców, który dodatkowo wykazał najwyższą skuteczność algorytmu 5. W celu uzyskania wyższych procentowych wyników testów, zaproponowano test łatwy, który wykazał, że każdy z algorytmów, przy odpowiednim doborze zbioru obrazów testowych, jest w stanie wykrywać punkty charakterystyczne z wysoką dokładnością. Wartość błędu granicznego została natomiast dobrana za pomocą testu błędu ludzkiego, w taki sposób, aby różne wskazania ekspertów mieściły się w obrębie błędu granicznego.

5.3.4. Przykładowe wyniki algorytmów

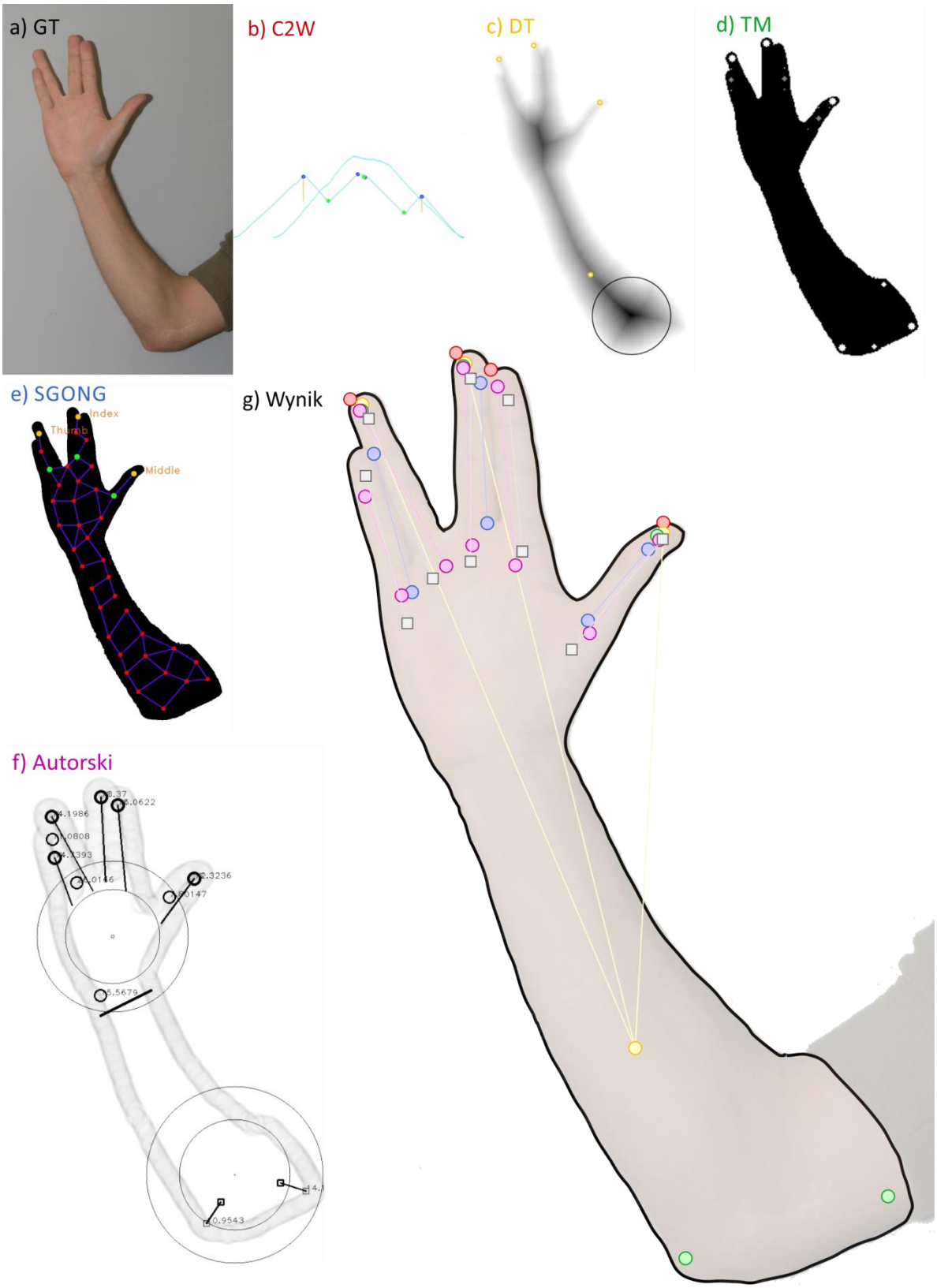
Niniejszy podrozdział poświęcony jest zbiorowej prezentacji wyników działania prezentowanych algorytmów. Na czterech zbiorczych rysunkach przedstawiono analizę wybranego gestu z użyciem opisywanych algorytmów, zgodnych z oznaczeniami w tabelach i wykresach.

Wybrane cztery gesty zostały wyłonione spośród bazy hgr2a. Pokrywają one typowe zachowania prezentowanych algorytmów. Są to gesty przedstawiające literę „A” oraz „O” Polskiego Alfabetu Palcowego, liczbę „3” Amerykańskiego Alfabetu Palcowego oraz znak specjalny w kształcie litery „V”.

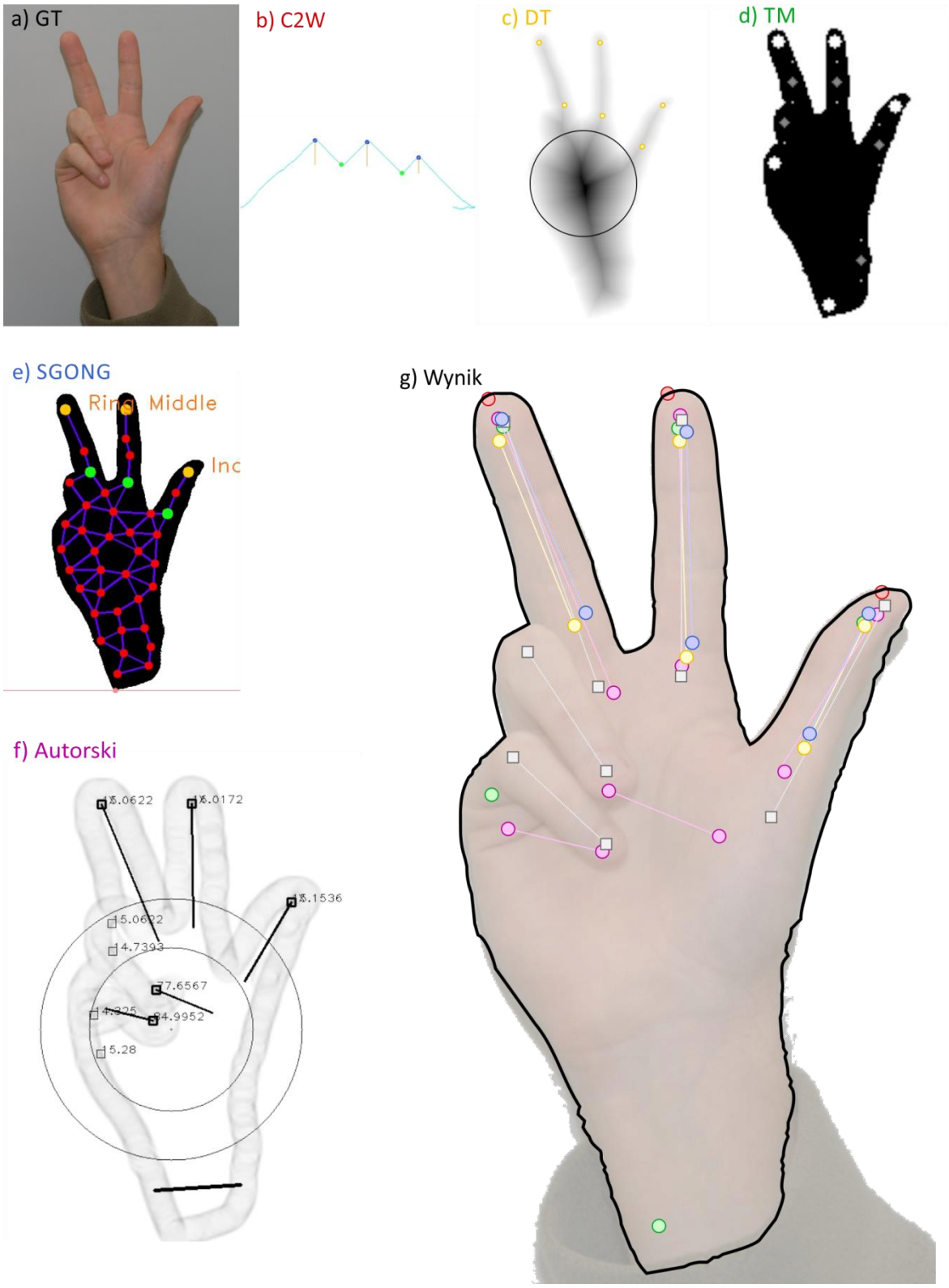
Rys. 55 przedstawia gest, w którym palce łączą się sprawiając, że mogą one wyglądać jakby były jednym grubym palcem. Rys. 56 oraz Rys. 57 są gestami zawierającymi łatwe do wykrycia odseparowane wyprostowane palce oraz dwa palce zgięte w specyficzny sposób. Rys. 58 przedstawia gest, w którym trudno sklasyfikować punkty charakterystyczne, ponieważ maska regionu skóry przedstawia zwartą bryłę

Dla algorytmu C2W, na przedstawionych rysunkach można zaobserwować dwie funkcję odległości punktów konturu od punktu nadgarstka dla obu części konturu po dwóch stronach nadgarstka. Lokalne maksima odwzorowywane są obrazach wynikowych jako czerwone koła na konturze. Sposób działania algorytmu powoduje, że punkty czubków palców mogą być wykryte jedynie na konturze. Wyprostowane i odseparowane palce wykrywane są poprawnie z dużą dokładnością, jednak w przypadku zwartych konturów i chowanych palców punkty są pomijane. Warto zwrócić uwagę na Rys. 55 1-b, gdzie mimo tego, że palce są połączone, lokalne wahania funkcji odległości są wystarczające aby odseparować palec wskazujący od środkowego.

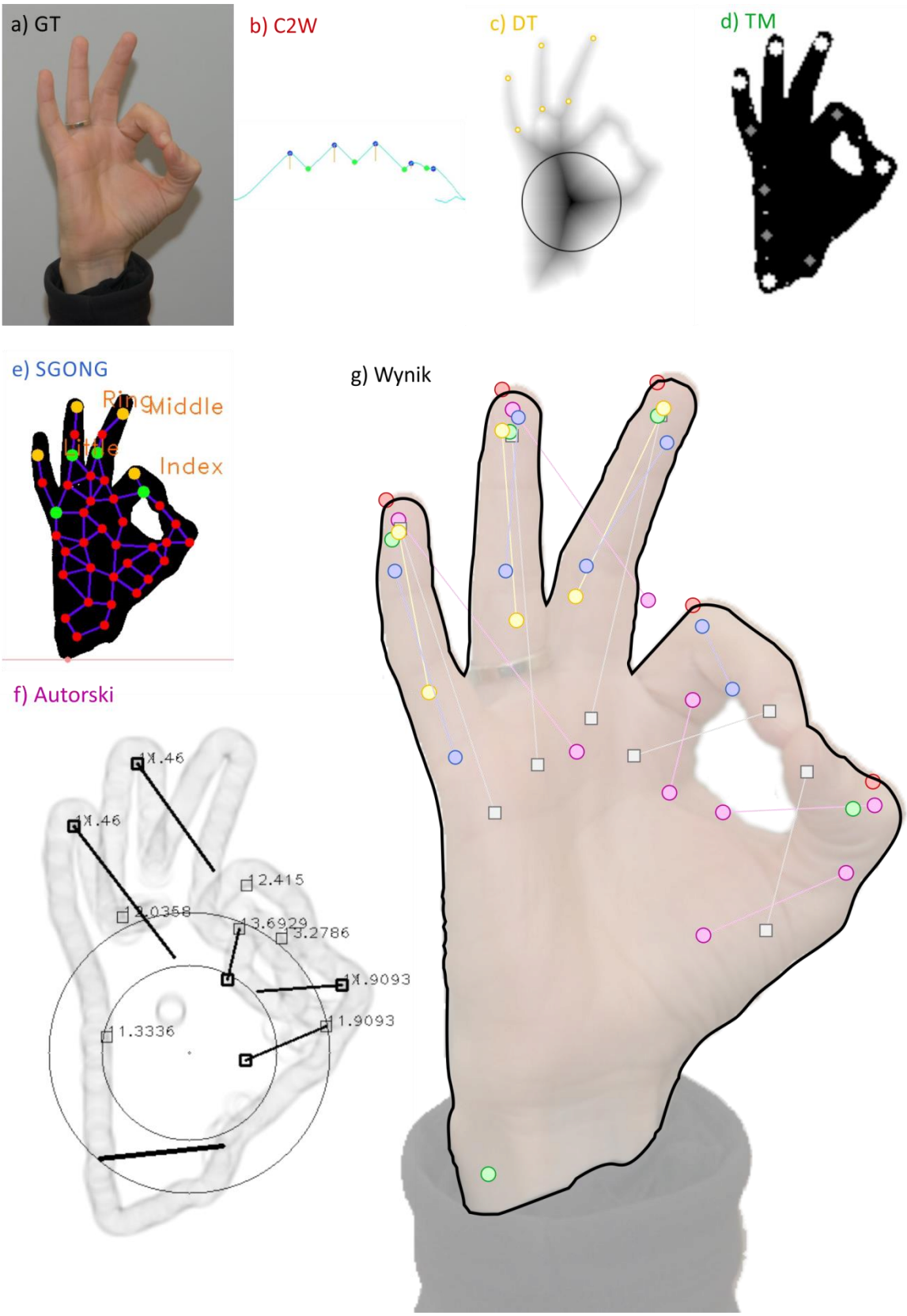
Algorytm DT przedstawiany jest na rysunkach w obrazach c). Sposób implementacji nie pozwala na łatwy dostęp do obrazów powstających w trakcie trwania algorytmu. Algorytm opiera się o analizę lokalnych maksimów transformaty odległościowej. Niestety algorytm nie działa poprawnie jeżeli region śródręcza nie jest poprawnie wykryty. Przykładem jest Rys. 58, gdzie region przedramienia wykryty został jako palec oraz Rys. 55 gdzie mimo wykrycia poprawnie czubków palców, nasada palców znaleziona została na przedramieniu.



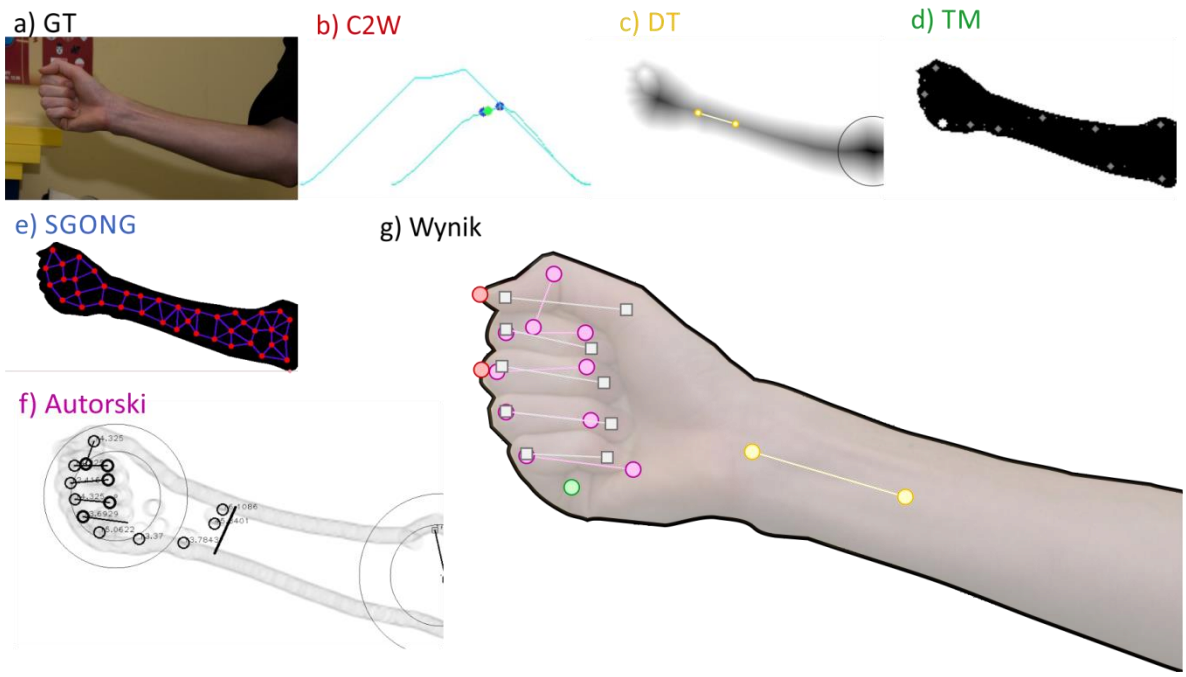
Rys. 55 a) Obraz wzorcowy przedstawiający gest specjalny w kształcie litery „V", b-f) przykłady obrazujące działanie algorytmów, g) naniesienie wykrytych punktów oraz punktów wzorcowych przedstawionych szarymi kwadratami.



Rys. 56 a) Obraz wzorcowy przedstawiający gest „3”, b-f) przykłady obrazujące działanie algorytmów, g) naniesienie wykrytych punktów oraz punktów wzorcowych przedstawionych szarymi kwadratami.



Rys. 57 a) Obraz wzorcowy przedstawiający gest „O”, b-f) przykłady obrazujące działanie algorytmów, g) naniesienie wykrytych punktów oraz punktów wzorcowych przedstawionych szarymi kwadratami.



Rys. 58 a) Obraz wzorcowy przedstawiający gest „A", b-f) przykłady obrazujące działanie algorytmów, g) naniesienie wykrytych punktów oraz punktów wzorcowych przedstawionych szarymi kwadratami.

Algorytm TM, wykrywa lokalne maksima mapy dopasowania wzorca. Wykrywanych jest 10 punktów. Punkty szare nie przeszły etapu filtracji fałszywych punktów. Spośród białych klasyfikowanych jest 5 najlepszych, ponieważ zakłada się, że tylko 5 palców widać na obrazie. Niestety, część punktów wykrywana jest na granicy konturu w regionie łokcia (zaobserwować to można na każdym obrazie). Poza dużą liczbą punktów fałszywych, algorytm dobrze wykrywa punkty odseparowane, oraz zdarza się, że wykrywa również punkty nie wykryte przez inne algorytmy, jak na przykład kciuk gestu z Rys. 57 3-a.

Algorytm SGONG przedstawiany jest na rysunkach w obrazach e). Założeniem algorytmu jest lokalizacja punktów w miejscach odseparowanych neuronów. Przedstawione rysunki pokazują poprawie wykryte punkty odseparowane. Jednak, aby zobaczyć typowe błędy tego algorytmu należy wrócić do Rys. 12.

Autorski algorytm wykazuje największą skuteczność w odwzorowaniu punktów charakterystycznych. Bez problemu wykrywa punkty charakterystyczne palców zgiętych przedstawianych na Rys. 58 oraz Rys. 55. Niestety, czasami procedura wykrywania punktów wymyka się z pod kontroli, odnajdując punkty całkowicie odbiegające od oczekiwań. Przykładem jest Rys. 56, gdzie nasada palca serdecznego wykryta została w miejscu nasady kciuka, oraz Rys. 57, na którym większość punktów została wykryta błędnie.

5.3.5. Wyznaczenie parametrów sterujących

Zgodnie z opisem z rozdziału 5.2.3, przeprowadzono procedurę doboru parametrów sterujących algorytmem. W celu wybrania odpowiednich wartości, maksymalizowano pole powierzchni pod wykresem dystrybuanty. W przybliżeniu wartość współczynnika jakości równa jest sumie wszystkich wartości funkcji dystrybuanty. Ponieważ funkcja jest dyskretna,

jej wartości liczone są przedziałowo. Dla przedstawionych przykładów, liczba przedziałów równa jest *N=21*.

Program sterowany jest parametrami, które zostały przedstawione w Tab. 13. Parametry zostały dobrane w taki sposób, aby maksymalizować wskaźnik jakości (24) oraz tak by miara testu F1 (23) była najwyższa.

Tab. 13 Zbiór parametrów sterujących algorytmem wyszukiwania punktów charakterystycznych dłoni

	Symbol	Wartość	Wpływ na czas	Opis
Poszukiwanie nadgarstka:	L	50	Duży	Ilość losowań par punktów jako kandydatów do najdłuższej przekątnej
	Δ	5	Mały	Dystans poprawiania lokalnego średnicy konturu
	β	0,66	Mały	Współczynnik kontrakcji przedziału
	δ	0,1	Mały	Krok poszukiwania maksimum
Wyznaczanie pozostałych punktów charakterystycznych:	F_r	3	Brak	Stosunek średnicy wzorca koła (palca) do wykrytej średnicy dłoni
	t_{DT}	0,2	Brak	Próg filtracji punktów zbyt blisko krawędzi
	F_N	9	Mały	Maksymalna liczba wykrywanych punktów czubków palców w obrazie
	τ	0,2	Mały	Parametr wyjścia z zapętlenia algorytmu poszukiwania nasady palca
	P_N	5	Mały	Maksymalna liczba wykrywanych regionów dłoni
	λ	1,4	Brak	Parametr proporcjonalnego powiększenia regionu śródręcza
	σ	1,5	Duży	Proporcjonalne powiększenie wzorca koła w celu ustalenia wielkość okna poszukiwania nasady palca
	t_E	0,001	Brak	Próg filtracji błędu E dla wektorów kierunków z obrazu kierunkowego

Wpływ zmian wartości parametrów na wynik detekcji oraz średni czas przetwarzania obrazu, jak i dokładny opis parametrów znajduje się w podpunktach poniżej.

- **Parametry metody poszukiwania nadgarstka**

Punkty nadgarstka odnajdywane są za pomocą algorytmu wyznaczenia najdłuższej średnicy spośród losowego zbioru punktów oraz z użyciem metody odnajdywania minimum wzdłuż kierunku z kontrakcją przedziału. Algorytm ten sterowany jest z użyciem 4 zmiennych.

Jeżeli po wykryciu pary punktów na konturze, L następnych losowo dobranych par nie tworzy dłuższej przekątnej, wtedy odcinek pomiędzy punktami uznawany jest za najdłuższą

przekątną. Sterowanie parametrem L zwiększa ilość iteracji potrzebnych, aby ostatecznie określić kąt obrotu obrazu. Kolejne zmienne zależne są od wyników wartości kąta, która wyznaczana jest za pomocą punktów losowo dobranych.

Jako, że punkty dobierane są losowo, ciężko przeprowadzić test, który dokładnie określiłby optymalną wartość parametrów. Problem analizy testu losowego przedstawiony jest w rozdziale 5.3.1, w którym przedstawiono jak zmieniają się wyniki testów przeprowadzanych wielokrotnie dla tych samych wartości parametrów. Dopiero uśrednione, wielokrotne analizy tego samego zbioru obrazów mogą doprowadzić do ustalenia optymalnych wartości parametrów. Jednak zysk na poprawie jakości jest zbyt niski, aby opłacało się przeprowadzać dogłębną analizę. Co więcej, zauważono, że drobne różnice w parametrach sterujących na tym etapie nie mają większego wpływu na jakość pozostałych punktów charakterystycznych. Ostatecznie postanowiono dobrać parametry intuicyjnie.

Skoro parametr L ma wpływ na ilość losowań, przebadano jego wpływ na czas działania algorytmu. Wyniki przedstawiono w Tab. 14. Zaobserwować można wyraźne zwiększenie średniego czasu analizy jednego obrazu wraz ze wzrostem wartości parametru. Dlatego parametr ten powinien mieć jak najmniejszą wartość. Jednocześnie, parametr ten musi mieć wysoką wartość aby zapewnić dostateczną ilość losowań potwierdzających słuszność wyboru danej pary punktów. Niestety, z powodu losowej natury testu, ciężko jest ustalić jednoznaczny wpływ parametru na wynik. Test dla L=50 został przeprowadzony 2 razy, wskazując dwa różne wyniki. Ostatecznie parametr ten został dobrany z wartością L=50.

Tab. 14 Wpływ parametru L na czas i jakość poszukiwania nadgarstka w obrazach z bazy hgr2a.

L:	Średni czas t[s]:	Ilość poprawnych punktów z błędem e<1.0:
30	0,0692	72,9%
50	0,0739	83,5%
	0,0741	76,4%
70	0,0793	78,8%

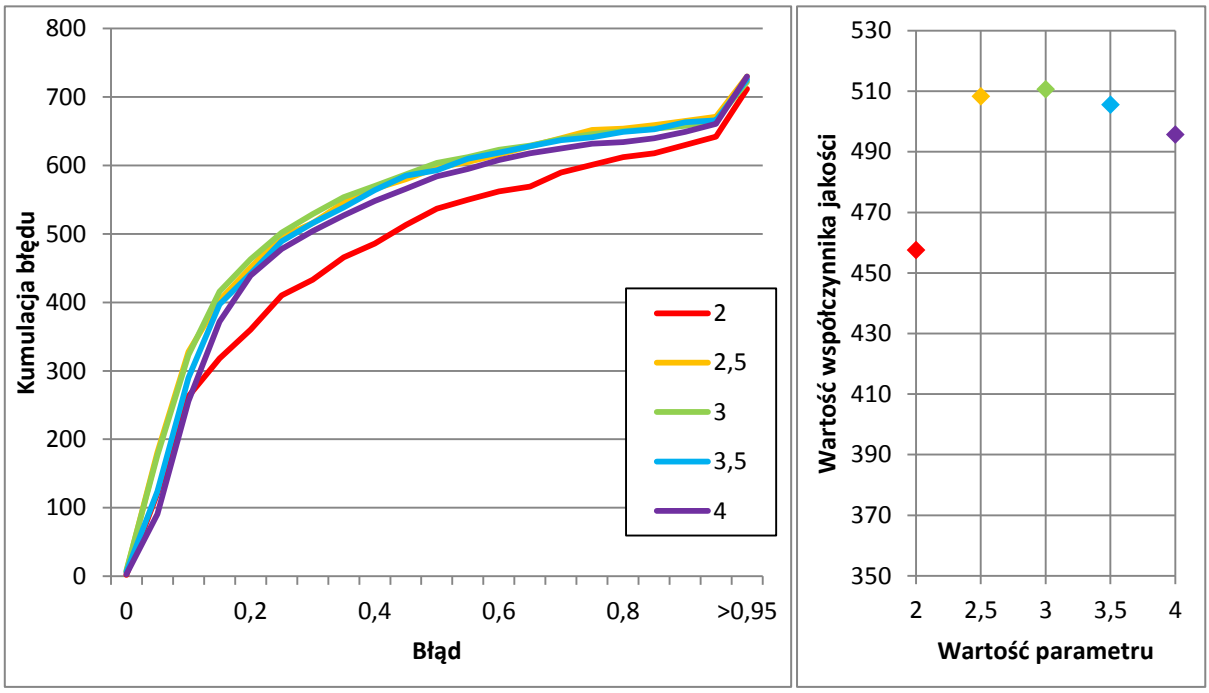
Wpływ kolejnych parametrów również uzależniony jest od wyników kąta dobranego losowo. Wszystkie trzy parametry nie mogą mieć zbyt niskiej wartości, ani zbyt wysokiej wartości. Zbyt niska wartość spowoduje dużą ilość kroków zanim osiągnięte zostanie ekstremum, a zbyt duża wartość może spowodować pominięcie punktu optymalnego. Dodatkowo parametry te mają pomijalny wpływ na czas analizy obrazu.

Parametr Δ określa o ile pikseli zostanie przesunięty punkt w ramach poprawiania lokalnego. Uznano za optymalne $\Delta=5$. Parametry δ i β są wartościami kroku początkowego oraz współczynnikiem kontrakcji przedziału. Długość kroku początkowego δ, ustalono jako 10% szerokości obrazu, dlatego $\delta=0,1$. Zgodnie z definicją, parametr β powinien znajdować się w przedziale (0,1), zatem uznano za optymalną wartość $\beta=0,66$.

Istnieje duża szansa, że przyjęte wartości nie są optymalne, jednak poszukiwanie optymalnych wartości jest czasochłonne, a otrzymane wyniki nie poprawią detekcji w znaczący sposób.

♦ **Stosunek średnicy wzorca koła (palca) do wykrytej średnicy dłoni - F_r**

Średnica dwóch wzorców koła i okręgu, które wykorzystywane są na etapie wyznaczania obrazów dopasowania wzorca, jest proporcjonalnie zależna od promienia regionu śródręcza. Pierwszym etapem algorytmu jest wyznaczenie regionu śródręcza jako największej wartości transformaty odległościowej. Dzieląc tę wartość przez współczynnik F_r otrzymywana jest średnica wzorca.



Rys. 59 Wpływ zmian parametru F_r na wykres kumulacji błędu i współczynnik jakości

Parametr ten ma duży wpływ na lokalizację wykrytych punktów. Powinien on być dobrany w taki sposób, aby wzorzec koła był takiej samej średnicy jak średnica palca. Zbyt mały lub zbyt duży wymiar wzorca sprawia, że odnajdywane punkty charakterystyczne nie znajdują się na czubkach palców. Jako, że wyznaczone punkty współczynnika jakości przybierają postać funkcji kwadratowej, wartość maksymalna została wyznaczona dla $F_r = 3$.

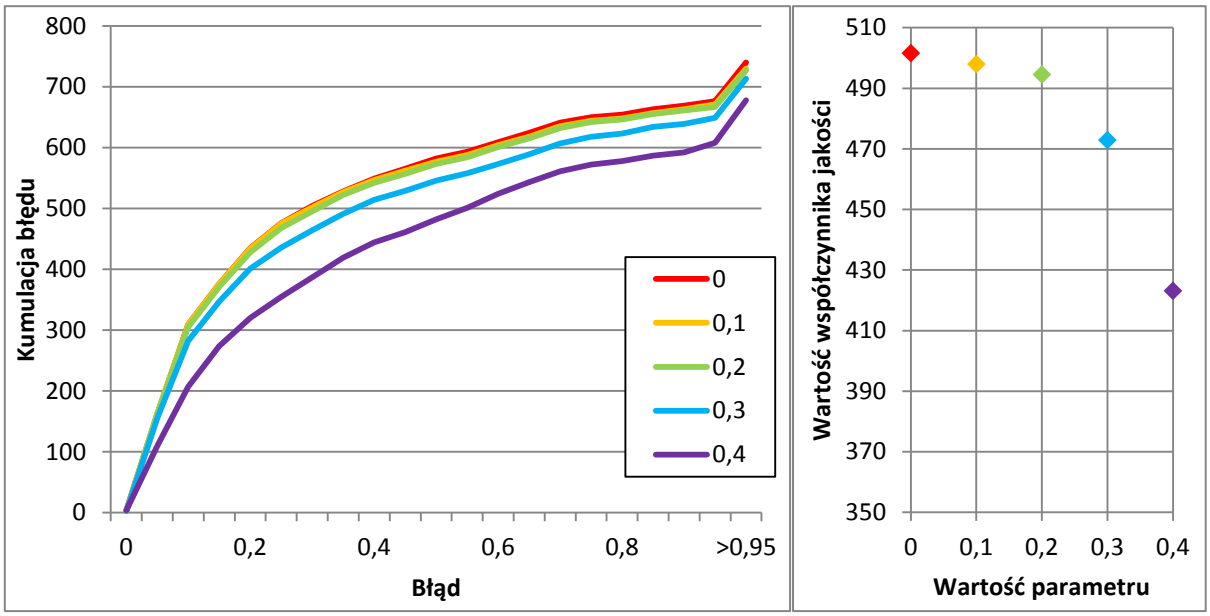
Parametr ten ma pomijalny wpływ na średni czas wykonywania algorytmu. Procedura dopasowania wzorca może działać wolniej w przypadku większych wzorców, jednak nie zauważono znaczącego spadku wydajności w trakcie analizy algorytmu.

Tab. 15 Wpływ zmian parametru F_r na wyniki testu

F_r	GT	Wy-kryte	Punkty bez pary	Sparowane punkty		Miary statystyczne			I
				TP	FP	TPR	PPV	F1	
2	826	732	116	431	279	78,79%	60,70%	68,58%	457,57
2,5	826	746	102	510	214	83,33%	70,44%	76,35%	508,33
3	826	734	112	521	193	82,31%	72,97%	77,36%	510,67
3,5	826	740	106	510	210	82,79%	70,83%	76,35%	505,57
4	826	748	99	501	226	83,50%	68,91%	75,51%	495,71

♦ **Próg filtracji punktów zbyt blisko krawędzi - t_{DT}**

Pierwsze kryterium filtracji wykrytych punktów polega na wyselekcjonowaniu punktów znajdujących się zbyt blisko krawędzi. W tym celu, dla każdego wykrytego punktu, pobierana jest wartość transformaty odległościowej o tych samych współrzędnych. W celu normalizacji obrazów o różnych wymiarach, współczynnik t_{DT} mnożony jest przez F_r . Innymi słowy, odrzucany jest punkt, który znajduje się w odległości proporcjonalnej do średnicy palca. Dla przykładu, jeżeli średnica palca w obrazie ma 20 pikseli, t_{DT} *=0,2 ,* wtedy jeżeli punkt zostanie wykryty z wartością transformaty odległościowej *DT=3*, czyli w odległości 3 pikseli od konturu, zostaje on odrzucony, ponieważ kryterium filtracji ustalone jest na *20 x 0,2=4.*



Rys. 60 Wpływ zmian parametru t_{DT} na wykres kumulacji i współczynnik jakości

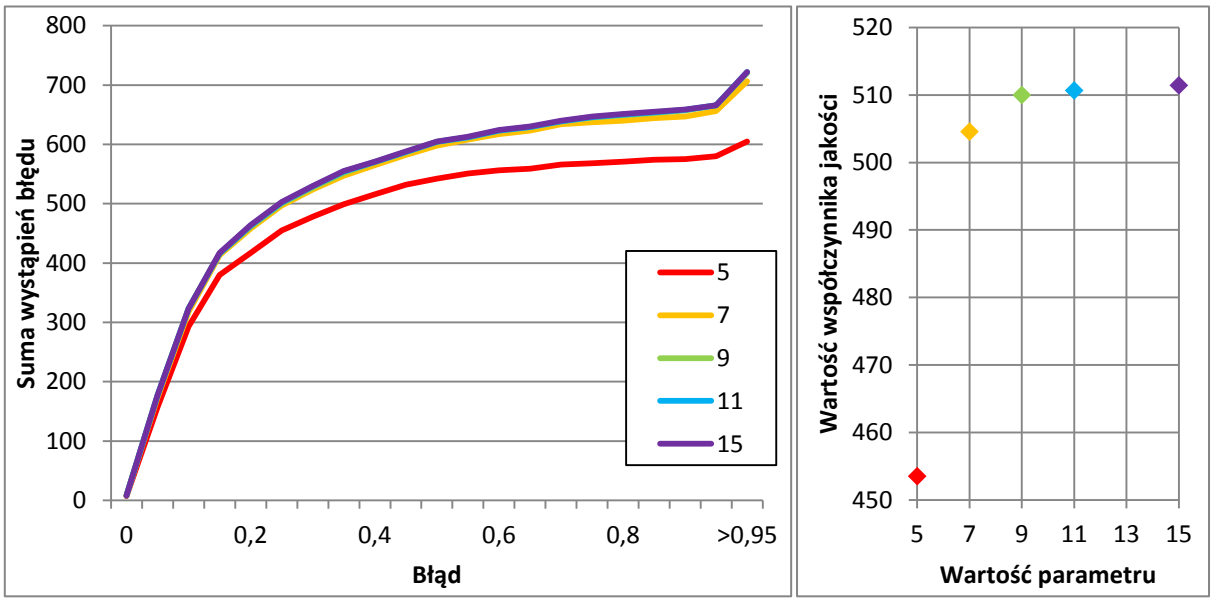
Tendencja spadkowa kolejnych wykresów kumulacji błędów dla parametru powoduje spadek współczynnika jakości wraz ze wzrostem parametru odrzutu. Jako, że współczynnika nie można dobrać na zasadzie wyznaczania maksimum funkcji regresji kwadratowej, należy podjąć inną metodę analizy. W celu wyznaczenia optymalnej wartości przeprowadzono analizę miar statystycznych, przedstawionych w Tab. 16. Wraz ze wzrostem parametru t_{DT}, selektywność spada, a precyzja rośnie. Kompromisem jest punkt 0,2, dla którego miara testu F1 przyjmuje maksimum.

Tab. 16 Wpływ zmian parametru t_{DT} na wyniki testu

t_{DT}	GT	Wy-kryte	Punkty bez pary	Sparowane punkty		Miary statystyczne			I
				TP	FP	TPR	PPV	F1	
0	826	758	90	500	236	84,75%	67,93%	75,41%	501,62
0,1	826	748	100	498	228	83,28%	68,60%	75,23%	498,00
0,2	826	746	102	501	223	83,08%	69,20%	75,51%	494,57
0,3	826	732	116	461	249	79,90%	64,93%	71,64%	472,90
0,4	826	692	152	383	291	71,59%	56,82%	63,36%	423,14

♦ **Maksymalna liczba wykrywanych punktów czubków palców w obrazie - F_N**

F_N jest to parametr określający kryterium stopu pętli poszukiwania punktów charakterystycznych klasyfikowanych jako czubki palców. Dla każdego obrazu wykrywane jest N potencjalnych palców, które następnie podlegają analizie.



Rys. 61 Wpływ zmian parametru F_N na wykres kumulacji i współczynnik jakości

Na etapie wykrywania palców nie jest określone, po której stronie nadgarstka należy szukać palców. Co więcej, ostateczny region śródręcza ustalany jest po tej stronie nadgarstka, po której wykryto więcej punktów potencjalnych palców. Dlatego istotne jest, aby parametr ten był nieparzysty oraz aby był w stanie pokryć wszystkie 5 palców dłoni, zakładając najgorszy przypadek, w którym dodatkowo 4 fałszywe punkty zostały wykryte po stronie łokcia.

Powyższa definicja parametru F_N wskazuje optymalną wartość równą 9. Analiza tego parametru, przedstawiona na Rys. 61 i w Tab. 17, wskazuje słuszność tego wyboru. Jeżeli punktów jest za dużo, wartość współczynnika jakości zostaje nasycona, a dodatkowe punkty są odrzucane. W przypadku zbyt niskiej wartości, nie wszystkie punkty zostają wykryte w obrazie.

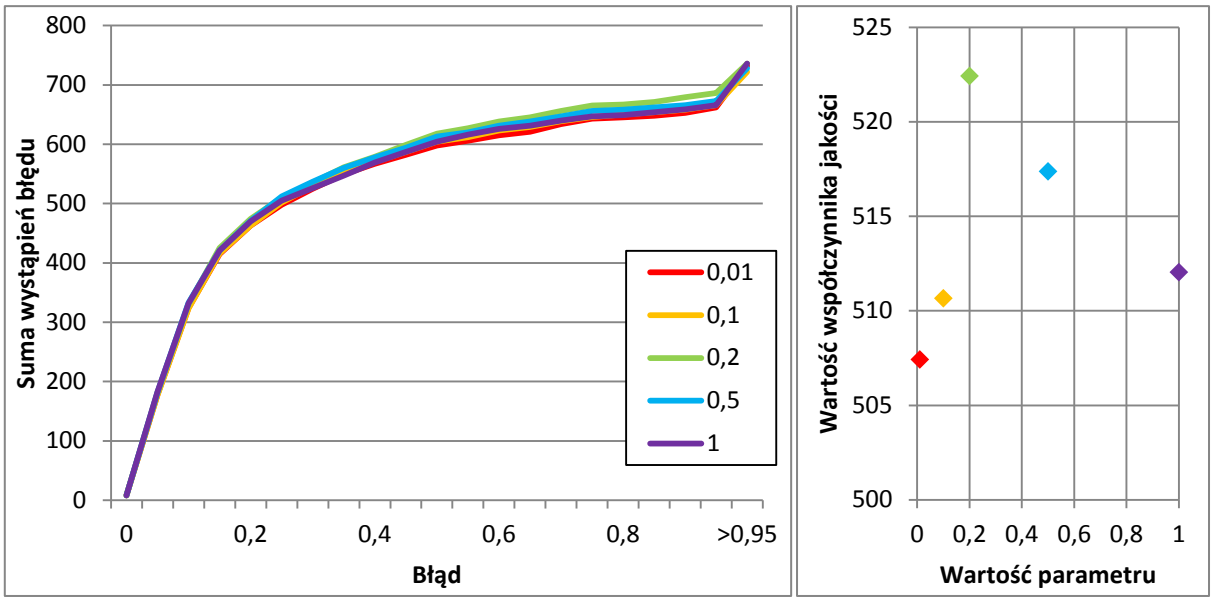
Parametr ma mały wpływ na czas działania algorytmu. W przypadku, gdy szukanych jest mniej punktów, algorytm analizuje obraz szybciej, ponieważ szybciej osiągane jest kryterium stopu.

Tab. 17 Wpływ zmian parametru F_N na wyniki testu

F_N	GT	Wy-kryte	Punkty bez pary	Sparowane punkty		Miary statystyczne			I
				TP	FP	TPR	PPV	F1	
5	826	600	228	471	127	67,38%	78,76%	72,63%	453,52
7	826	714	128	516	182	80,12%	73,93%	76,90%	504,57
9	826	732	114	519	193	81,99%	72,89%	77,17%	510,00
11	826	734	112	521	193	82,31%	72,97%	77,36%	510,67
15	826	736	112	522	192	82,33%	73,11%	77,45%	511,43

♦ **Parametr wyjścia z zapętlenia algorytmu poszukiwania nasady palca - τ**

W trakcie etapu poszukiwania nasady palca, algorytm przemieszcza się po maksimach transformaty odległościowej. Celem jest wyznaczenie ścieżki wzdłuż palca, od czubka do nasady. W przypadku, gdy w obrazie kierunkowym pojawi się szum wewnątrz palca, powstanie lokalne minimum transformaty odległościowej, a algorytm osiągnie kryterium stopu zbyt wcześnie. W takich przypadkach, jeżeli długość palca jest zbyt mała, a nasada jest wykryta zbyt daleko od regionu śródręcza, algorytm wykonuje procedurę porzucenia lokalnego minimum. W takim przypadku ścieżka generowana jest prostolinijnie w kierunku środka śródręcza z krokiem o długości równym odległości punktu od środka śródręcza, pomnożonego przez parametr τ.



Rys. 62 Wpływ zmian parametru τ na wykres kumulacji i współczynnik jakości

Wykres współczynnika jakości można aproksymować kwadratową funkcją regresyjną. Wartość maksymalna występuje dla punktu 0,2. Jednak różnice we współczynniku jakości nie są duże, a wykresy kumulacji błędu prawie się pokrywają. Oznacza to, że przypadków błędnego zapętlenia w trakcie wykrywania nasady palca jest mało. Co więcej, jeżeli występują, to częściej w przypadku poszukiwania palców zgiętych, ponieważ wartość dla parametru 1 ma większą wartość indeksu jakości niż dla parametru 0,01. Wniosek ten wywodzi się z faktu, że wykryte czubki palców zgiętych w regionie środka śródręcza mają mniejszy błąd dopasowania, niż punkty nasady palców w regionie środka śródręcza.

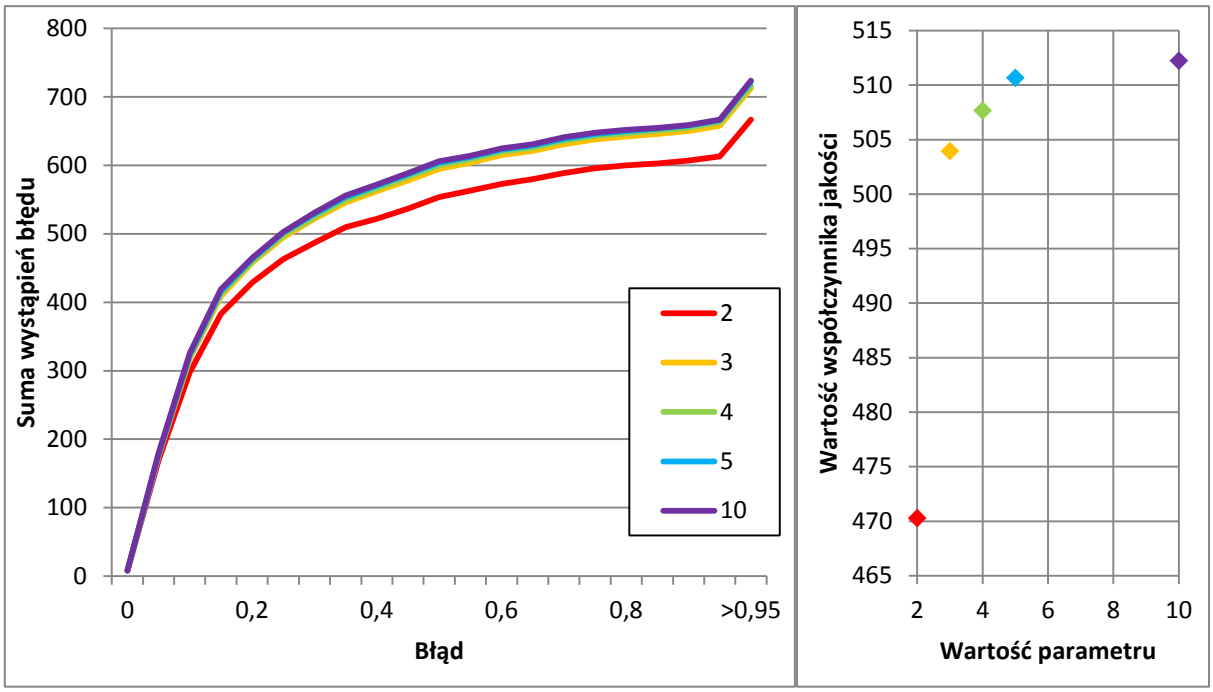
Tab. 18 Wpływ zmian parametru τ na wyniki testu

τ	GT	Wy-kryte	Punkty bez pary	Sparowane punkty		Miary statystyczne			I
				TP	FP	TPR	PPV	F1	
0,01	826	744	102	517	207	83,52%	71,41%	76,99%	507,43
0,1	826	734	112	521	193	82,31%	72,97%	77,36%	510,67
0,2	826	748	98	527	201	84,32%	72,39%	77,90%	522,43
0,5	826	740	106	529	198	83,31%	72,76%	77,68%	517,38
1	826	748	98	518	210	84,09%	71,15%	77,08%	512,05

♦ **Maksymalna liczba wykrywanych regionów dłoni - P_N**

W przypadku użytkownika o masywnie zbudowanej ręce, z odsłoniętym łokciem lub krótkim rękawem, maksimum globalne transformaty odległościowej zostanie wykryte na przedramieniu. Co więcej, nie tylko pierwsze maksimum globalne, ale również kolejne maksima globalne mogą zostać wykryte wzdłuż regionu przedramienia. Celem wykrywania kilku potencjalnych regionów śródręcza jest odnalezienie ich po obu stronach nadgarstka.

Mógłby się zatem nasuwać wniosek, że lepiej wykryć tylko dwóch kandydatów po obu stronach nadgarstka, jednak czasem dla przypadków obrazów dłoni z długim rękawem, obszar drugiej strony nadgarstka jest tak pomijalnie mały, że jego analiza jest zbędna, a gdy wszystkie punkty zostaną wykryte po jednej stronie, czas realizacji algorytmu będzie krótszy.



Rys. 63 Wpływ zmian parametru P_N na wykres kumulacji i współczynnik jakości

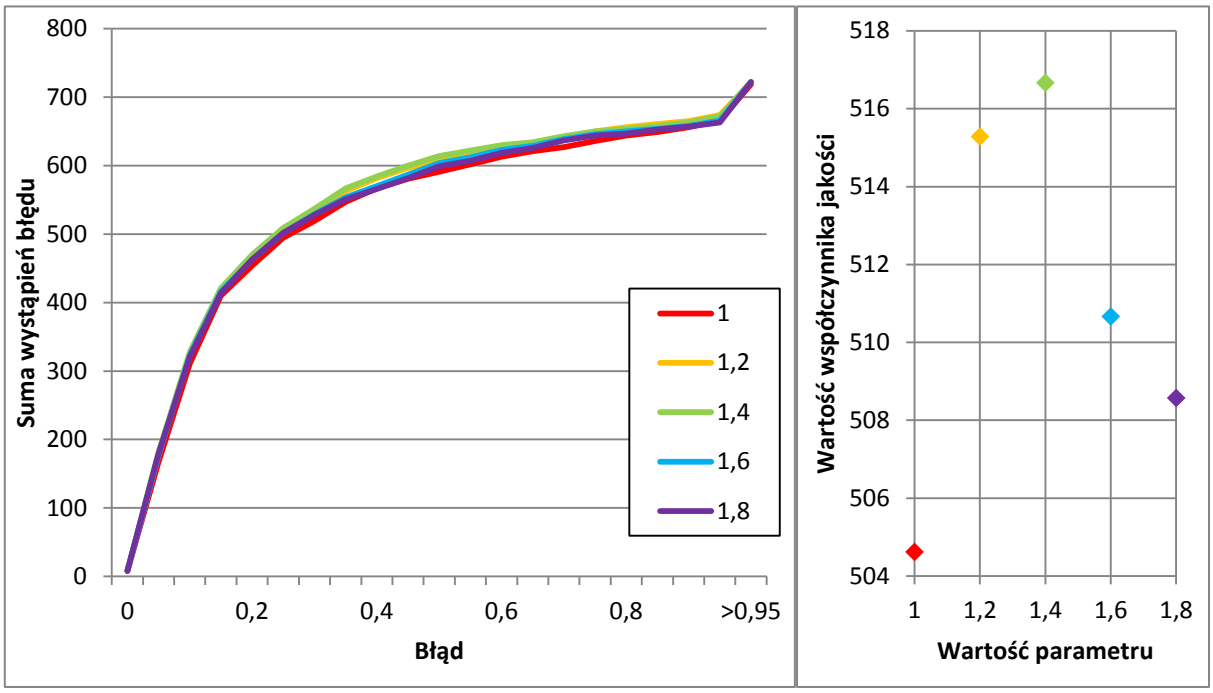
Badania wykazują tendencję stabilizującą się po parametrze równym 5. Dalszy przyrost parametru negatywnie wpływa na czas realizacji algorytmu, ponieważ więcej potencjalnych kandydatów musi być odnalezionych, mimo, że i tak zostaną oni odrzuceni. Tendencję stabilizującą można zaobserwować analizując miary statystyczne przedstawione w Tab. 19.

Tab. 19 Wpływ zmian parametru P_N na wyniki testu

P_N	GT	Wy-kryte	Punkty bez pary	Sparowane punkty		Miary statystyczne			I
				TP	FP	TPR	PPV	F1	
2	826	670	167	479	180	74,15%	72,69%	73,41%	470,29
3	826	722	121	514	191	80,94%	72,91%	76,72%	503,95
4	826	728	118	517	191	81,42%	73,02%	76,99%	507,67
5	826	734	112	521	193	82,31%	72,97%	77,36%	510,67
10	826	736	110	523	193	82,62%	73,04%	77,54%	512,24

♦ **Parametr proporcjonalnego powiększenia regionu śródręcza - λ**

W celu określenia regionu przynależności wykrytych punktów do klasy palców zgiętych lub wyprostowanych, region śródręcza powiększany jest proporcjonalnie o współczynnik λ. Punkty wykryte po różnych stronach powiększonego okręgu śródręcza mają inną wartość oczekiwaną długości palca oraz inne kryterium stopu dla odnalezienia punktu nasady palca.



Rys. 64 Wpływ zmian parametru λ na wykres kumulacji i współczynnik jakości

Jako, że algorytmy poszukiwania palców zgiętych i wyprostowanych nieznacznie się różnią, a parametr ma jedynie wpływ na lokalizację punktów nasady palców, wartość współczynnika jakości nieznacznie się zmienia dla różnych wartości parametru λ. Analiza wykresu i aproksymacji regresją funkcji kwadratowej wykazała wartość maksimum w punkcie *1,4*. Dodatkowo w Tab. 20 zauważyć można, że parametr ten wpływa tylko na selektywność testu, mając pomijalny wpływ na precyzję.

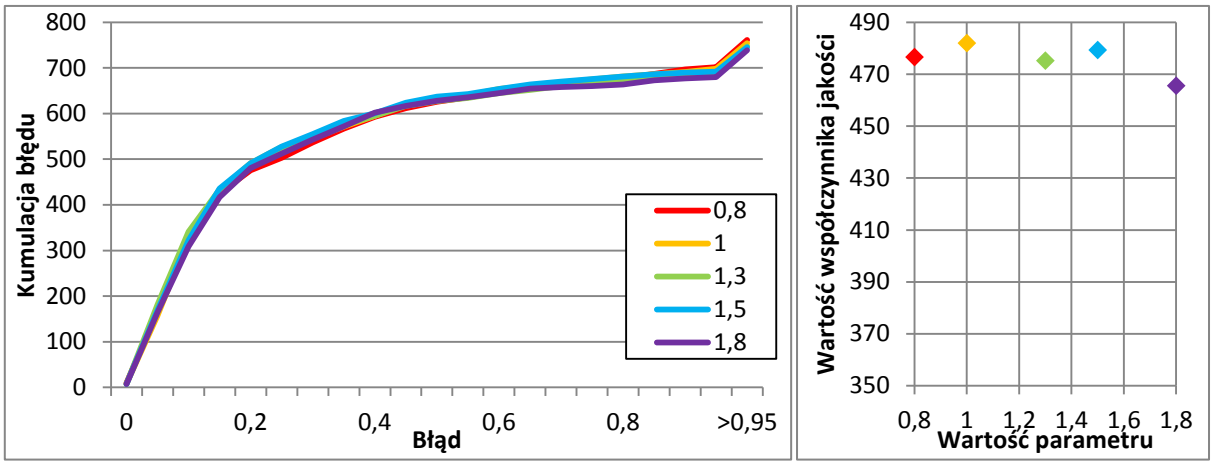
Parametr ten nie wpływa na czas wykonywania programu, ponieważ używany jest jedynie do klasyfikacji punktów wykrytych jako wartość progowa.

Tab. 20 Wpływ zmian parametru λ na wyniki testu

λ	GT	Wy-kryte	Punkty bez pary	Sparowane punkty		Miary statystyczne			I
				TP	FP	TPR	PPV	F1	
1	826	731	115	511	200	81,63%	71,87%	76,44%	504,62
1,2	826	734	112	526	188	82,45%	73,67%	77,81%	515,29
1,4	826	734	112	529	185	82,53%	74,09%	78,08%	516,67
1,6	826	734	112	521	193	82,31%	72,97%	77,36%	510,67
1,8	826	734	112	520	194	82,28%	72,83%	77,27%	508,57

♦ **Proporcjonalne powiększenie wzorca koła - σ**

Parametr wykorzystywany jest do ustalenia wielkości okna na etapie poszukiwania ścieżki palca od czubka do nasady, wędrując po maksimach transformaty odległościowej. Po wykryciu maksimum z transformaty odległościowej, na region maksimum lokalnego nanoszone jest koło o promieniu równym wartości tego maksimum. Oznacza to, że następne maksimum będzie szukane w pewnej odległości od obecnego maksimum. Odległość ta określana jest przez szerokość okna poszukiwania σ. Dla parametru równego 1, maksima szukane są na obrzeżu zamazanego obszaru. Dla parametru mniejszego niż 1, praca odbywa się przez ciągłe wykonywanie procedury porzucania lokalnego minimum z użyciem parametru τ. Dopiero dla parametru o wartości powyżej 1 algorytm działa poprawnie. Dodatkowo, dla parametru o wartości 2 lub więcej, występować będą efekty przeskakiwania ścieżki z jednego palca na inny, ponieważ region poszukiwania będzie dwukrotnie większy niż grubość palca.



Rys. 65 Wpływ zmian parametru σ na wykres kumulacji i współczynnik jakości

Z wykresu można zaobserwować dwa regiony działania, dla wartości mniejszej lub równej 1 oraz dla wartości większych. Wartość współczynnika jakości dla strefy pierwszej jest wyższa, jednak jest to strefa działania niepożądanego. Świadczy o tym też niska selektywność, przedstawiona w Tab. 21. Biorąc pod uwagę wszystkie miary statystyczne, optymalną wartością jest σ=1,5.

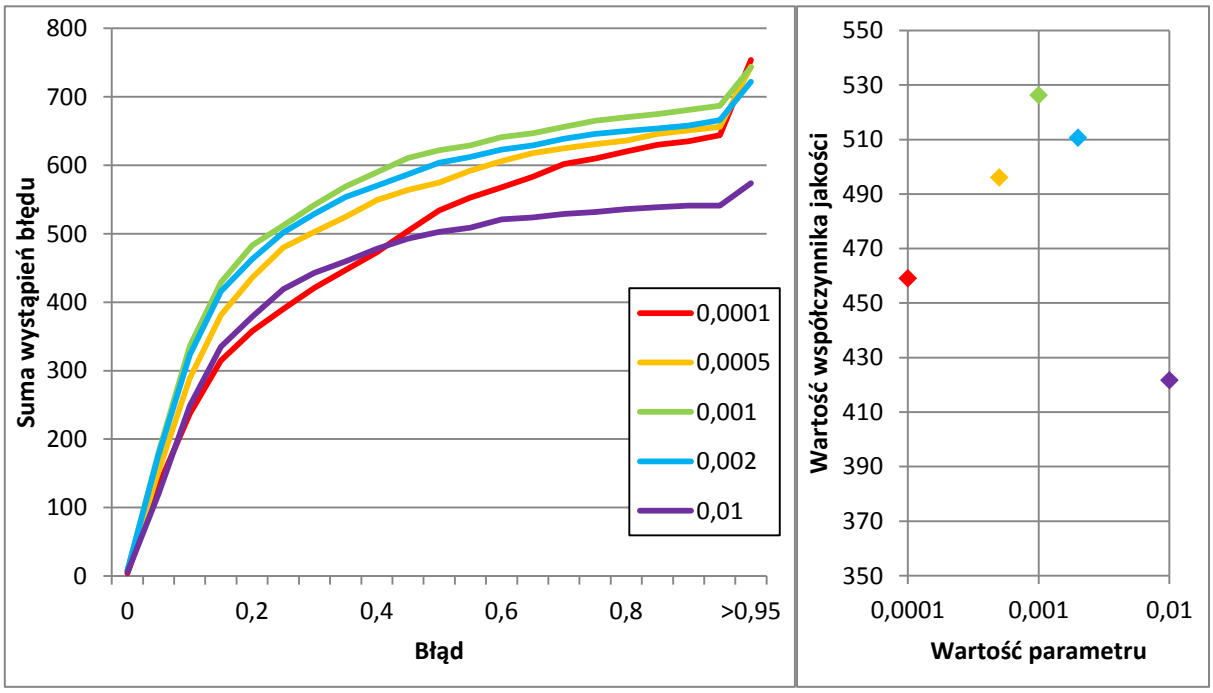
Dodatkowo, ten etap jest najbardziej czasochłonny. Oznacza to, że im większy krok jest wykonywany, tym szybciej, jednak mniej dokładniej osiągnięty zostanie punkt nasady. Analiza średniego czasu wykonywania algorytmu dla obrazu $t_{śr}$ przedstawiona jest w Tab. 21.

Tab. 21 Wpływ zmian parametru σ na wyniki testu

σ	GT	Wy-kryte	Punkty bez pary	Spar. punkty		Miary statystyczne			I	$t_{śr}$ [s]
				TP	FP	TPR	PPV	F1		
0,8	826	776	74	528	224	87,71%	70,21%	77,99%	530,19	0,289
1,0	826	770	80	538	208	87,06%	72,12%	78,89%	534,43	0,270
1,3	826	764	86	540	200	86,26%	72,97%	79,06%	530,19	0,260
1,5	826	762	88	548	190	86,16%	74,25%	79,77%	536,05	0,253
1,8	826	754	94	536	196	85,08%	73,22%	78,71%	526,62	0,234

♦ **Próg filtracji błędu E dla wektorów kierunków z obrazu kierunkowego - t_E**

Obrazowy opis wpływu tego parametru przedstawia Rys. 23. Obraz kierunkowy rysowany jest za pomocą odcinków o odpowiednim kierunku oraz z wystarczająco dużym natężeniem gradientu. Parametr t_E określa próg tego natężenia. Dla niższych wartości obraz dokładnie odwzorowuje strukturę skóry dłoni ze wszystkimi zmarszczkami. Im wartość jest wyższa tym mniej niepotrzebnych informacji zawartych jest w obrazie, jednak dla zbyt dużych wartości tracone są informacje o konturach palców zgiętych.



Rys. 66 Wpływ zmian parametru t_E na wykres kumulacji i współczynnik jakości

Wartość współczynnika jakości można aproksymować kwadratową funkcją regresji. Dodatkowo parametr ten, ze względu na swoją strukturę, został przedstawiony na skali logarytmicznej. Widać, że dla parametru o wartości 0,001 najwięcej szczegółów zostało odwzorowanych, zapewniając wysoką detekcję punktów wewnątrz maski, jednocześnie nie wzmacniając niepotrzebnych szczegółów.

Proces progowania nie ma wpływu na czas wykonywania algorytmu.

Tab. 22 Wpływ zmian parametru t_E na wyniki testu

t_E	GT	Wy-kryte	Punkty bez pary	Sparowane punkty		Miary statystyczne			I
				TP	FP	TPR	PPV	F1	
2	826	772	76	417	333	84,58%	55,60%	67,10%	459,10
2,5	826	766	84	496	246	85,52%	66,85%	75,04%	496,10
3	826	760	90	535	201	85,60%	72,69%	78,62%	526,29
3,5	826	734	112	521	193	82,31%	72,97%	77,36%	510,67
4	826	570	257	438	131	63,02%	76,98%	69,30%	421,71

6. Podsumowanie i kierunki dalszych prac

Niniejsza rozprawa doktorska dotyczy budowy systemu, w którym zaimplementowano nowy algorytm wykrywania i lokalizacji punktów charakterystycznych dłoni w obrazach cyfrowych. Dziedzina tematu pochodzi z obszaru rozpoznawania gestów. Pierwszy rozdział poświęcony jest przeglądowi literatury. Spośród opisywanych zagadnień, wybrano prace skupiające się na wykrywaniu gestów wykonywanych za pomocą ułożenia dłoni. W celu rozpoznania takich gestów, przebadano algorytmy lokalizujące punkty charakterystyczne dłoni, takie jak punkty nadgarstka, czubki i nasady palców.

Postawiono cel oraz ograniczenia ostatecznego produktu, jakimi są: przetwarzanie obrazu w czasie rzeczywistym, brak kosztownego i specjalistycznego sprzętu, uniwersalność i brak odgórnych założeń środowiskowych, takich jak długość rękawa, orientacja dłoni, kontrolowane tło i oświetlenie. Spowodowało to, że ostateczny system został stworzony w oparciu o tradycyjne obrazy cyfrowe z użyciem metod przetwarzania obrazu.

Na etapie przeglądu literatury wybrano cztery wiodące algorytmy, które opierały się na metodach przetwarzania obrazu w celu ekstrakcji cech charakterystycznych. Zaimplementowano te algorytmy we wspólnym środowisku testowym. Środowisko to zawierało 3 duże bazy obrazów, które składały się z obrazów dłoni w różnym ułożeniu (gestów Polskiego i Amerykańskiego Języka Migowego), maski występowania regionów skóry oraz pliki XML ze współrzędnymi punktów charakterystycznych oznaczanych przez ekspertów.

Zaproponowano również autorski algorytm sztucznej inteligencji, z użyciem obrazu kierunkowego oraz zbiór metod przetwarzania obrazu. Zaproponowano zbiór reguł decyzyjnych, które przedstawiono w postaci drzewa decyzyjnego. Wykryte lokalne punkty charakterystyczne przetwarzanych obrazów poddawane były klasyfikacji, która określała przynależność danego punktu do grupy punktów palców lub punktów fałszywie wykrytych. Ostatecznie, od punktów czubków palców prowadzona była ścieżka poszukiwania punktów nasady palców, aby stworzyć pełny opis dłoni za pomocą punktów charakterystycznych. Algorytm ten dostrajany był za pomocą zbioru parametrów. Parametry te dobierane były w taki sposób, aby maksymalizować zaproponowany wskaźnik jakości.

Algorytm ostatecznie porównano z wybranymi algorytmami wybranymi w trakcie przeglądu literatury. Jako, że algorytm łączył w sobie metody poszukiwania dwóch rodzajów punktów, czyli odnajdywania punktów nadgarstka oraz punktów palców, oba algorytmy zostały przetestowane oddzielnie i porównane z wybranymi algorytmami z literatury.

Przedstawiono metodę walidacyjną, która opierała się na analizie błędu dopasowania par punktów charakterystycznych wykrytych oraz wzorcowych. Błąd był zaproponowany jako miara euklidesowa pary punktów. Idealnie odnaleziony punkt znajdował się na pozycji punktu wskazanego przez eksperta, a wzajemna odległość tej pary punktów wynosiła 0. Walidację przeprowadzono za pomocą wykresów dystrybuanty oraz miar statystycznych oceny klasyfikatorów, wyliczanych z użyciem liczby punktów poprawnie wykrytych, nieprawnie wykrytych oraz punktów niewykrytych.

Ostateczne wyniki testów pokazały skuteczność zaproponowanego algorytmu, który cechował się dużą selektywnością oraz nieodstającą od pozostałych algorytmów precyzją. Oznacza to, że zaproponowany algorytm, mimo niepoprawienia w znaczącym stopniu precyzji, wykrywał znacznie więcej punktów charakterystycznych, dzięki czemu ostateczny opis dłoni był pełniejszy. Wykazuje to ostatecznie poprawność postawionej tezy, że wykorzystane w pracy metody sztucznej inteligencji pozwalają na dokładniejszą lokalizację i detekcję wybranych punktów charakterystycznych dłoni.

W przyszłości, algorytm może zostać rozszerzony o śledzenie cech charakterystycznych na sekwencjach wideo, w celu identyfikacji gestów dynamicznych. Dodatkowo planowane jest odwzorowywanie wykrytych punktów charakterystycznych w celu tworzenia pełnych modeli dłoni, tak aby możliwe było łatwe określenie gestu, lub dedukcja lokalizacji palców, które nie są widoczne, lub nie zostały wykryte. Ponadto pomocne może być stworzenie kaskady detektorów, czyli analizowanie jednego obrazu z użyciem wielu metod, a następnie łączenia wyników tak, aby uzyskać jak najlepsze rezultaty.

Planuje się również zaproponowanie konkretnego praktycznego zastosowania oraz opracowanie do tego celu słusznego klasyfikatora gestów. Przykładami, w których możliwe jest wykorzystanie wyżej wymienionych badań, są między innymi systemy wspierające komunikację osób głuchoniemych, interaktywne kontrolery środowiska rozszerzonej rzeczywistości lub programy sterujące robotami, na przykład w sterylnym otoczeniu.

Dodatkowo, należy zauważyć, że w myśl podziału opisanego we wstępie do tej pracy, gesty dłoni są najmniejszym elementem zbioru gestów należących do komunikacji niewerbalnej. Posiadając algorytm realizujący zadanie najniższego poziomu, można wykorzystać jego możliwości do algorytmów wyższego poziomu, jak na przykład odwzorowanie całej sylwetki człowieka.

Bibliografia

[1] W. J. MacLean, R. Herpers, C. Pantofaru, L. Wood, K. Derpanis, D. Topalovic, and J. K. Tsotsos, "Fast hand gesture recognition for real-time teleconferencing applications," in *Proc. of the workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS) in conjunction with the IEEE International Conference on Computer Vision (ICCV)*, 2001.

[2] J. Wachs, H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith, and J. Handler, "A real-time hand gesture interface for medical visualization applications," in *Applications of Soft Computing*, ser. Advances in Intelligent and Soft Computing, A. Tiwari, R. Roy, J. Knowles, E. Avineri, and K. Dahal, Eds. Springer Berlin Heidelberg, 2006, vol. 36, pp. 153–162.

[3] M. A. Al-Mouhamed, O. Toker, and A.-K. Al-Harthy, "A 3-d vision-based man-machine interface for hand-controlled telerobot." *IEEE Transactions on Industrial Electronics*, vol. 52, no. 1, pp. 306–319, 2005.

[4] Y. Shen, S. K. Ong, and A. Y. C. Nee, "Vision-based hand interaction in augmented reality environment," *International Journal of Human-Computer Interaction*, vol. 27, no. 6, pp. 523–544, 2011.

[5] D. Kim, J. Lee, H.-S. Yoon, J. Kim, and J. Sohn, "Vision-based arm gesture recognition for a long-range human–robot interaction," *The Journal of Supercomputing*, vol. 65, no. 1, pp. 336–352, 2013.

[6] R. Bastos and M. Sales Dias, "Skin color profile capture for scale and rotation invariant hand gesture recognition," in *Gesture-Based Human-Computer Interaction and Simulation*, ser. Lecture Notes in Computer Science, M. Sales Dias, S. Gibet, M. Wanderley, and R. Bastos, Eds. Springer Berlin Heidelberg, 2009, vol. 5085, pp. 81–92.

[7] T. Grzejszczak, M. Mikulski, T. Szkodny, and K. Jędrasiak, "Gesture based robot control," in *Computer Vision and Graphics*, ser. Lecture comments in Computer Science, L. Bolc, R. Tadeusiewicz, L. Chmielewski, and K. Wojciechowski, Eds. Springer Berlin Heidelberg, 2012, vol. 7594, pp. 407–413.

[8] S. Mitra and T. Acharya, "Gesture recognition: A survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 3, pp. 311–324, May 2007.

[9] J.-F. Collumeau, R. Leconge, B. Emile, and H. Laurent, "Hand-gesture recognition: Comparative study of global, semi-local and local approaches," in *Image and Signal Processing and Analysis (ISPA), 2011 7th International Symposium on*, Sept 2011, pp. 247–252.

[10] N. Iwane, "Arm movement recognition for flag signaling with kinect sensor," in *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2012 IEEE International Conference on*. IEEE, 2012, pp. 86–90.

[11] N. Nguyen-Duc-Thanh, D. Stonier, S. Lee, and D.-H. Kim, "A new approach for human-robot interaction using human body language," in *Convergence and Hybrid Information Technology*, ser. Lecture comments in Computer Science, G. Lee, D. Howard, and D. Ślęzak, Eds. Springer Berlin Heidelberg, 2011, vol. 6935, pp. 762–769.

[12] A. M. Hrisca, "The silent language of an artificial body," *Acta Universitatis Danubius. Communicatio*, vol. 6, no. 1, pp. 5 – 20, 2012.

[13] T. Pfister, X. Li, G. Zhao, and M. Pietikainen, "Recognising spontaneous facial micro-expressions," in *2011 IEEE International Conference on Computer Vision (ICCV)*, Nov 2011, pp. 1449–1456.

[14] K. Wolf, A. Naumann, M. Rohs, and J. Müller, "A taxonomy of microinteractions: Defining microgestures based on ergonomic and scenario-dependent requirements," in *Human-Computer Interaction – INTERACT 2011*, ser. Lecture comments in Computer Science, P. Campos, N. Graham, J. Jorge, N. Nunes, P. Palanque, and M. Winckler, Eds. Springer Berlin Heidelberg, 2011, vol. 6946, pp. 559–575.

[15] C. Endres, T. Schwartz, and C. A. Muller, "Geremin: 2d microgestures for drivers based on electric field sensing," in *Proceedings of the 16th international conference on Intelligent user interfaces*. ACM, 2011, pp. 327–330.

[16] A. Riener and P. Wintersberger, "Natural, intuitive finger based input as substitution for traditional vehicle control," in *Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, ser. AutomotiveUI '11. New York, NY, USA: ACM, 2011, pp. 159–166.

[17] W. Hürst and C. van Wezel, "Gesture-based interaction via finger tracking for mobile augmented reality," *Multimedia Tools and Applications*, vol. 62, no. 1, pp. 233–258, 2013.

[18] F. Tafazzoli and R. Safabakhsh, "Model-based human gait recognition using leg and arm movements," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 8, pp. 1237 – 1246, 2010.

[19] A. Karaśkiewicz, "Niewerbalne elementy komunikacji w sytuacjach oficjalnych. analiza debaty z udziałem liderów politycznych," *INVESTIGATIONES LINGUISTICAE*, vol. 20, pp. 23 – 32, 2010.

[20] J. Perlin and B. Szczepankowski, *Polski język migowy: opis lingwistyczny*. Wydawn. Szkolne i Pedagogiczne, 1992.

[21] W. C. Stokoe, D. C. Casterline, and C. G. Croneberg, *A dictionary of American Sign Language on linguistic principles*. Linstok Press Silver Spring, 1976.

[22] N. Tanibata, N. Shimada, and Y. Shirai, "Extraction of hand features for recognition of sign language words," in *In International Conference on Vision Interface*, 2002, pp. 391–398.

[23] B. Szczepankowski, *Niesłyszący, głusi, głuchoniemi: wyrównywanie szans*. Wydawnictwa Szkolne i Pedagogiczne-Spółka Akcyjna, 1999.

[24] W.-P. Lee, C. Kaoli, and J.-Y. Huang, "A smart tv system with body-gesture control, tag-based rating and context-aware recommendation," *Knowledge-Based Systems*, vol. 56, no. 0, pp. 167 – 178, 2014.

[25] A. Bochenek and M. Reicher, *Anatomia człowieka. Tom 1*. Warszawa: Wydawnictwo Lekarskie PZWL, 2012.

[26] A. JOHN and A. MUSIOLIK, "Analiza kinematyczna palców ręki," *Modelowanie Inżynierskie*, vol. 9, pp. 111–116, 2010.

[27] Microsoft. Kinect. [Online]. Available: http://www.xbox.com/pl-PL/kinect/

[28] Samsung. Telewizor – serce domowego centrum rozrywki. [Online]. Available: http://www.samsung.com/pl/article/a-home-entertainment-center-inside-your-tv

[29] Samsung galaxy s4 – funkcja air gesture. [Online]. Available: http://www.samsung.com/pl/article/a-home-entertainment-center-inside-your-tv

[30] Google. Trends. [Online]. Available: http://www.google.com/trends/explore#q=gesture%2C%20kinekt&cmpt=q

[31] P. Halarnkar, S. Shah, H. Shah, H. Shah, and J. Shah, "Gesture recognition technology: A review," *International Journal of Engineering Science and Technology*, vol. 4, no. 11, pp. 4648 – 4654, 2012.

[32] O. Luzanin and M. Plancak, "Hand gesture recognition using low-budget data glove and cluster-trained probabilistic neural network," *Assembly Automation*, vol. 34, no. 1, pp. 94–105, 2014.

[33] F. Camastra and D. De Felice, "Lvq-based hand gesture recognition using a data glove," in *Neural Nets and Surroundings*, ser. Smart Innovation, Systems and Technologies, B. Apolloni, S. Bassis, A. Esposito, and F. C. Morabito, Eds. Springer Berlin Heidelberg, 2013, vol. 19, pp. 159–168.

[34] G. Lu, L.-K. Shark, G. Hall, and U. Zeshan, "Immersive manipulation of virtual objects through glove-based hand gesture interaction," *Virtual Reality*, vol. 16, no. 3, pp. 243–252, 2012.

[35] M. Ishikawa and H. Matsumura, "Recognition of a hand-gesture based on self-organization using a dataglove," in *Neural Information Processing, 1999. Proceedings. ICONIP '99. 6th International Conference on*, vol. 2, 1999, pp. 739–745 vol.2.

[36] S. Iba, J. Weghe, C. J. J. Paredis, and P. Khosla, "An architecture for gesture-based control of mobile robots," in *Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on*, vol. 2, 1999, pp. 851–857 vol.2.

[37] C. Lee and Y. Xu, "Online, interactive learning of gestures for human/robot interfaces," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 4, Apr 1996, pp. 2982–2987 vol.4.

[38] L. Jing, Y. Zhou, Z. Cheng, and T. Huang, "Magic ring: A finger-worn device for multiple appliances control using static finger gestures," *Sensors*, vol. 12, no. 5, pp. 5775–5790, 2012.

[39] Y. Takeoka, T. Miyaki, and J. Rekimoto, "Z-touch: An infrastructure for 3d gesture interaction in the proximity of tabletop surfaces," in *ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '10. New York, NY, USA: ACM, 2010, pp. 91–94.

[40] A. Scoditti, "Gestural interaction techniques for handheld devices combining accelerometers and multipoint touch screens," Ph.D. dissertation, Universit'e de Grenoble, 2011.

[41] L. Lamberti and F. Camastra, "Handy: A real-time three color glove-based gesture recognizer with learning vector quantization," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10489 – 10494, 2012.

[42] S. Raghupatruni, N. Nasam, and K. Lingam, "Article: Sixth sense enabled campus - possibilities and challenges," *International Journal of Computer Applications*, vol. 75, no. 8, pp. 39–44, August 2013, full text available.

[43] Sony. Kontroler ruchu playstationmove. [Online]. Available: http://pl.playstation.com/ps3/accessories/detail/-item303966/Kontroler-ruchu-PlayStation%C2%AEMove/

[44] T. Krzeszowski, B. Kwolek, and K. Wojciechowski, "Model-based 3d human motion capture using global-local particle swarm optimizations," in *Computer Recognition Systems 4*, ser. Advances in Intelligent and Soft Computing, R. Burduk, M. Kurzyński, M. Woźniak, and A. Żołnierek, Eds. Springer Berlin Heidelberg, 2011, vol. 95, pp. 297–306.

[45] M. Tang, “Recognizing hand gestures with microsoft’s kinect,” *Palo Alto: Department of Electrical Engineering of Stanford University*, pp. 1 – 7, 2011.

[46] K. K. Biswas and S. Basu, “Gesture recognition using microsoft kinect;,” in *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*, Dec 2011, pp. 100–103.

[47] Y. Li, “Hand gesture recognition using kinect,” in *Software Engineering and Service Science (ICSESS), 2012 IEEE 3rd International Conference on*, June 2012, pp. 196–199.

[48] G. Hackenberg, R. McCall, and W. Broll, “Lightweight palm and finger tracking for real-time 3d gesture control,” in *Virtual Reality Conference (VR), 2011 IEEE*, March 2011, pp. 19–26.

[49] D. Droeschel, J. Stuckler, and S. Behnke, “Learning to interpret pointing gestures with a time-of-flight camera,” in *Proceedings of the 6th International Conference on Human-robot Interaction*, ser. HRI ’11. New York, NY, USA: ACM, 2011, pp. 481–488.

[50] M. Van den Bergh and L. Van Gool, “Combining rgb and tof cameras for real-time 3d hand gesture interaction,” in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, Jan 2011, pp. 66–72.

[51] A. Sahami Shirazi, Y. Abdelrahman, N. Henze, S. Schneegass, M. Khalilbeigi, and A. Schmidt, “Exploiting thermal reflection for interactive systems,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’14. New York, NY, USA: ACM, 2014, pp. 3483–3492.

[52] E. Larson, G. Cohn, S. Gupta, X. Ren, B. Harrison, D. Fox, and S. Patel, “Heatwave: Thermal imaging for surface user interaction,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’11. New York, NY, USA: ACM, 2011, pp. 2565–2574.

[53] P. Mistry and P. Maes, “Mouseless: A computer mouse as small as invisible,” in *CHI ’11 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA ’11. New York, NY, USA: ACM, 2011, pp. 1099–1104.

[54] W. Malina and M. Smiatacz, *Cyfrowe przetwarzanie obrazów*. Akademicka Oficyna Wydawnicza EXIT, 2008.

[55] R. Tadeusiewicz and P. Korohoda, *Komputerowa analiza i przetwarzanie obrazów*. Wydawnictwo Fundacji Postępu Telekomunikacji, 1997.

[56] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, “A survey of skin-color modeling and detection methods,” *Pattern Recognition*, vol. 40, no. 3, pp. 1106 – 1122, 2007.

[57] M. Kawulok, “Fast propagation-based skin regions segmentation in color images,” in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, April 2013, pp. 1–7.

[58] M. Kawulok, J. Kawulok, and J. Nalepa, “Spatial-based skin detection using discriminative skin-presence features,” *Pattern Recognition Letters*, vol. 41, no. 0, pp. 3 – 13, 2014.

[59] M. F. Abdelkader, W. Abd-Almageed, A. Srivastava, and R. Chellappa, “Silhouette-based gesture and action recognition via modeling trajectories on riemannian shape manifolds,” *Computer Vision and Image Understanding*, vol. 115, no. 3, pp. 439 – 455, 2011.

[60] C.-B. Park and S.-W. Lee, “Real-time 3d pointing gesture recognition for mobile robots with cascade hmm and particle filter,” *Image and Vision Computing*, vol. 29, no. 1, pp. 51 – 63, 2011.

[61] S. Martin, C. Tran, A. Tawari, J. Kwan, and M. Trivedi, “Optical flow based head movement and gesture analysis in automotive environment,” in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, Sept 2012, pp. 882–887.

[62] M. Holte, T. Moeslund, and P. Fihl, “View-invariant gesture recognition using 3d optical flow and harmonic motion context,” *Computer Vision and Image Understanding*, vol. 114, no. 12, pp. 1353 – 1361, 2010.

[63] J. Bao, A. Song, Y. Guo, and H. Tang, “Dynamic hand gesture recognition based on surf tracking,” in *Electric Information and Control Engineering (ICEICE), 2011 International Conference on*, April 2011, pp. 338–341.

[64] Y. Yao and C.-T. Li, “A framework for real-time hand gesture recognition in uncontrolled environments with partition matrix model based on hidden conditional random fields,” in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, Oct 2013, pp. 1205–1210.

[65] P. Barros, N. Júnior, J. Bisneto, B. Fernandes, B. Bezerra, and S. Fernandes, “An effective dynamic gesture recognition system based on the feature vector reduction for surf and lcs,” in *Artificial Neural Networks and Machine Learning – ICANN 2013*, ser. Lecture comments in Computer Science, V. Mladenov, P. Koprinkova-Hristova, G. Palm, A. Villa, B. Appollini, and N. Kasabov, Eds. Springer Berlin Heidelberg, 2013, vol. 8131, pp. 412–419.

[66] C.-C. Lien and C.-L. Huang, “Model-based articulated hand motion tracking for gesture recognition,” *Image and Vision Computing*, vol. 16, no. 2, pp. 121 – 134, 1998.

[67] N. Shimada, Y. Shirai, Y. Kuno, and J. Miura, “Hand gesture estimation and model refinement using monocular camera-ambiguity limitation by inequality constraints,” in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, Apr 1998, pp. 268–273.

[68] C.-C. Lien and C.-L. Huang, “Model-based articulated hand motion tracking for gesture recognition,” *Image and Vision Computing*, vol. 16, no. 2, pp. 121–134, 1998.

[69] E. Stergiopoulou, K. Sgouropoulos, N. Nikolaou, N. Papamarkos, and N. Mitianoudis, “Real time hand detection in a complex background,” *Engineering Applications of Artificial Intelligence*, vol. 35, no. 0, pp. 54 – 70, 2014.

[70] Z. Feng, B. Yang, Y. Chen, Y. Zheng, T. Xu, Y. Li, T. Xu, and D. Zhu, “Features extraction from hand images based on new detection operators,” *Pattern Recognition*, vol. 44, no. 5, pp. 1089 – 1105, 2011.

[71] Q. Chen, N. D. Georganas, and E. Petriu, “Real-time vision-based hand gesture recognition using haar-like features,” in *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, May 2007, pp. 1–6.

[72] S. Malik and J. Laszlo, “Visual touchpad: A two-handed gestural input device,” in *Proceedings of the 6th International Conference on Multimodal Interfaces*, ser. ICMI ’04. New York, NY, USA: ACM, 2004, pp. 289–296.

[73] T. Grzejszczak, A. Gałuszka, M. Niezabitowski, and K. Radlak, “Comparison of hand feature points detection methods,” in *Technological Innovation for Collective Awareness Systems*. Springer Berlin Heidelberg, 2014, pp. 167–174.

[74] E. Stergiopoulou and N. Papamarkos, “Hand gesture recognition using a neural network shape fitting technique.” *Eng. Appl. of AI*, vol. 22, no. 8, pp. 1141–1158, 2009.

[75] A. Licsár and T. Szirányi, “Hand gesture recognition in camera-projector system.” in *ECCV Workshop on HCI*, ser. Lecture comments in Computer Science, N. Sebe, M. S. Lew, and T. S. Huang, Eds., vol. 3058. Springer, 2004, pp. 83–93.

[76] T. Grzejszczak, J. Nalepa, and M. Kawulok, “EnglishReal-time wrist localization in hand silhouettes,” in *EnglishProceedings of the 8th International Conference on Computer Recognition Systems CORES 2013*, ser. Advances in Intelligent Systems and Computing, R. Burduk, K. Jackowski, M. Kurzynski, M. Wozniak, and A. Zolnierek, Eds. Springer International Publishing, 2013, vol. 226, pp. 439–449.

[77] J. Nalepa, T. Grzejszczak, and M. Kawulok, “Wrist localization in color images for hand gesture recognition,” in *Man-Machine Interactions 3*, ser. Advances in Intelligent Systems and Computing, D. A. Gruca, T. Czachórski, and S. Kozielski, Eds. Springer International Publishing, 2014, vol. 242, pp. 79–86.

[78] Y. Sato, Y. Kobayashi, and H. Koike, “Fast tracking of hands and fingertips in infrared images for augmented desk interface,” in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, 2000, pp. 462–467.

[79] L. Dung and M. Mizukawa, “Fast hand feature extraction based on connected component labeling, distance transform and hough transform,” *Journal of Robotics and Mechatronics*, vol. Vol.21, No.6, pp. 726–738, 2009.

[80] M. Czupryna, “Human–computer interaction using hand gestures,” Master’s thesis, Politechnika Śląska, 2011.

[81] M. Czupryna and M. Kawulok, “Real-time vision pointer interface,” in *ELMAR, 2012 Proceedings*, Sept 2012, pp. 49–52.

[82] S. Russell and P. Norvig, *Artificial Intelligence. A modern approach*. Prentice-Hall, Egnlewood Cliffs, 1995, vol. 25.

[83] L. Bownik, *Sieć Semantyczna: Reprezentacja i Logika*. Katowice : Instytut Technik Innowacyjnych EMAG, 2009.

[84] J. Wilson Robin, *Wprowadzenie do teorii grafów*. Wydawnictwo Naukowe PWN, 2007.

[85] G. Bradski, “Opencv library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[86] G. Borgefors, “Distance transformations in digital images,” *Computer vision, graphics, and image processing*, vol. 34, no. 3, pp. 344–371, 1986.

[87] Y. Wang, M. Shi, and T. Wu, “A method of fast and robust for traffic sign recognition,” in *Image and Graphics, 2009. ICIG ’09. Fifth International Conference on*, Sept 2009, pp. 891–895.

[88] J. Bohm, A. S. Frangakis, R. Hegerl, S. Nickell, D. Typke, and W. Baumeister, “Toward detecting and identifying macromolecules in a cellular context: template matching applied to electron tomograms,” *Proceedings of the National Academy of Sciences*, vol. 97, no. 26, pp. 14245–14250, 2000.

[89] M. Kawulok and J. Szymanek, “Precise multi-level face detector for advanced analysis of facial images,” *Image Processing, IET*, vol. 6, no. 2, pp. 95–103, March 2012.

[90] T. Grzejszczak, “Semantic representation of block world environment: algorithm of scene reasoning from incomplete information,” *Przegląd Elektrotechniczny*, vol. 87, pp. 201–204, 2011.

[91] T. Grzejszczak and A. Galuszka, “On planning in multi-agent environment: algorithm of scene reasoning from incomplete information,” in *Advances in Computational Intelligence*. Springer, 2011, pp. 613–620.

[92] T. Grzejszczak, “Objects recognition on block world environment scene: Algorithm of objects detection and classification,” *XIII International PHD Workshop OWD*, vol. 29, pp. 287–290, 2011.

[93] B. Bukowiecki, A. Galuszka, and S. Szweda, “Optimization of the geometrical features of the powered roof support unit kinematic chain due to the criterion of minimizing the weight,” *Przegląd Elektrotechniczny*, vol. 88, no. 3 A, pp. 131–138, 2012.

[94] A. Agnihotri, B. Purwar, N. Jeebun, and S. Agnihotri, “Determination of sex by hand dimensions,” *The Internet Journal of Forensic Science*, vol. 1, no. 2, pp. 12–24, 2006.

Spis ilustracji

Rys. 1 Podział gestów ze względu na obszar zainteresowania 8
Rys. 2 Przykłady gestów statycznych oraz dynamicznych. 10
Rys. 3 Oznaczenie punktów charakterystycznych dłoni: czerwony – czubek palca, opuszka; błękitny – więzadła; niebieski – punkty oraz region nadgarstka; zielony – region śródręcza; żółty – punkty wklęsłe między palcami. 12
Rys. 4 Przykłady ulokowania punktów charakterystycznych 13
Rys. 5 Wykresy popularności wyszukiwania słów "gesture" (niebieska linia) oraz "kinekt"(czerwona linia) z użyciem wyszukiwarki Google, stworzone przez Google Trends [30]. Po prawej zaprezentowano frazy, w których najczęściej pojawiało się słowo „gesture”. 15
Rys. 6 Przykład mapy odległościowej wygenerowanej przez Kinect 18
Rys. 7 Przykład działania algorytmu analizy odległości punktów konturu od nadgarstka. a) Kontur obrazu źródłowego, b) funkcja odległości kolejnych punktów konturu od punktu nadgarstka 23
Rys. 8 Przykład wyznaczenia punktów typu AB (pomarańczowy), oraz środka dłoni (zielony) i orientacji dłoni (fioletowy). Rysunek pobrany z publikacji autora [80] 25
Rys. 9 Ostateczne wyznaczenie punktów charakterystycznych, nasady oraz czubków palców na podstawie punktów typu C. Rysunek pobrany z publikacji autora [80] 26
Rys. 10 Przykład działania algorytmu. a) wzorzec, b) maska z wykrytymi punktami, c) określenie regionu poszukiwań. Rysunek pobrany z publikacji autora [78] 27
Rys. 11 Przykład wyniku końcowego algorytmu SGONG. a) Obraz pobrany z publikacji autora [74]. b) Obraz uzyskany po zaimplementowaniu algorytmu od podstaw. 31
Rys. 12 Przykłady częstych błędów algorytmu SGONG 32
Rys. 13 a) Obraz wejściowy reprezentujący rękę oraz tło, b) maska dłoni wskazana przez eksperta, c) mapa prawdopodobieństwa występowania regionu skóry. 35
Rys. 14 Przykład działania transformacji odległościowej typu L2. Po lewej fragment jednobitowej maski. Po prawej transformata odległościowa w postaci macierzy liczb rzeczywistych, w którym maksymalna wartość reprezentowana jest jako kolor biały. 37
Rys. 15 Przykład działania algorytmu dopasowania wzorca koła o wymiarach 5×5 do maski dłoni o wymiarach 100×100. Pokazano wyniki splotu dla przykładowych 4 punktów. Obraz docelowy o wymiarach 95×95 przedstawia wyniki splotu każdego punktu. 38
Rys. 16 Kolejne etapy uzyskiwania obrazu kierunkowego. b) Zbliżenie wskazanego na obrazie źródłowym a), regionu wraz z wizualizacją wyliczenia gradientu dla 3 wybranych pikseli. c) Wyznaczenie kąta stycznej dla co trzeciego piksela w co trzecim rzędzie, korzystając z 25 gradientów. Czerwona oraz niebieska strzałka zostaje zwizualizowana jako odcinek na obrazie kierunkowym d). W macierzy 5×5 oznaczonej żółtym kolorem, wariacja jest zbyt wysoka, co powoduje brak wizualizacji odcinka w tych współrzędnych. 40
Rys. 17 a-d) Przykłady konturów utworzonych z masek dłoni z zaznaczonymi odcinkami nadgarstka oraz orientacji dłoni. e) Przykład konturu utworzonego z maski nie spełniającego założeń. 42
Rys. 18 Poprawianie lokalne etapu poszukiwania najdłuższej przekątnej. Kolorem niebieskim oznaczone są potencjalne nowe przekątne, a czerwonym odcinki nie spełniające warunku zawierania się wewnątrz maski. 44
Rys. 19 Kolejne etapy wykrywania punktów nadgarstka. a) Odnalezienie najdłuższej cięciwy konturu, d) wyznaczenie funkcji profilu obróconego obrazu b), d) naniesienie wykrytych punktów na obraz oryginalny. 45
Rys. 20 Proces orientacji wykrytych punktów: a) obracanie obrazu zgodnie z kątem nachylenia najdłuższej cięciwy PQ, b) obracanie punktów nadgarstka UV nanosząc je w odpowiednie miejsca w obrazie pierwotnym 47

Rys. 21 Sposób pozyskania profilu maski dłoni z zaznaczonym globalnym maksimum (czerwony) oraz kilkoma lokalnymi minimami (zielony). a) Metoda polegająca na odnalezieniu najlepszego z minimów lokalnych, b) metoda filtracji profilu 47
Rys. 22 Przykład z bazy gestów. a) Obraz barwny przedstawiający gest „3", b) maska z regionem skóry. 49
Rys. 23 a) Obraz kierunkowy wewnątrz maski dla obrazu kolorowego, b) Obraz kierunkowy progowany..... 50
Rys. 24 c) Wynik dopasowania a) wzorca okręgu do obrazu kierunkowego (Rys. 23 b). d) Wynik dopasowania b) wzorca koła do maski dłoni (Rys. 22 b). 50
Rys. 25 Połączenie obu map prawdopodobieństwa dopasowania wzorca. Czerwonymi punktami zaznaczono przykładowe miejsca wykrycia maksimów lokalnych. 51
Rys. 26 Wynik transformacji odległościowej a) dla obrazu kierunkowego, b) dla maski dłoni. c) Analiza transformaty odległościowej maski dłoni w celu odnalezienia regionu śródręcza. 52
Rys. 27 Pełna procedura algorytmu autorskiego..... 53
Rys. 28 Przykłady konsekwencji wykrycia nadgarstka a) poprawnie oraz b) błędnie. Czerwone punkty odrzucane są, ponieważ znajdują się zbyt blisko niebieskiego odcinka nadgarstka. 54
Rys. 29 Podział stref klasyfikacji wykrytych punktów charakterystycznych. Czerwony – Strefa odrzutu punktów zbyt blisko konturu; Zielony/niebieski – Podział na regiony dłoni i przedramienia poprzez linię nadgarstka; Jasny kolor – Punkty klasyfikowane jako palce wystające; Ciemny kolor – Punkty klasyfikowane jako palce zgięte; Koła symbolizują strefy śródręcza i powiększone strefy śródręcza..... 56
Rys. 30 Przykład wykrycia punktu czubka palca wyprostowanego (żółty krzyżyk) i punktu nasady palca (zielony krzyżyk). a) Punkt wykryty jest na Rys. 25, b) Rys. 26-a służy do wykrycia ścieżki palca..... 57
Rys. 31 Przykład wykrycia punktu palca zgiętego (żółty krzyżyk) i punktu nasady palca (zielony krzyżyk), która aproksymowana jest na końcu danego regionu. a) Punkt wykryty jest na Rys. 25, b) Rys. 26-a służy do wykrycia ścieżki palca. 57
Rys. 32 Przykład wykrycia punktu palca zgiętego (żółty krzyżyk) i wnioskowania nasady palca (zielony krzyżyk), która aproksymowana jest na postawie odległości. a) Punkt wykryty jest na Rys. 25, b) Rys. 26-a służy do wykrycia ścieżki palca. 58
Rys. 33 Drzewo decyzyjne klasyfikacji wykrytych punktów charakterystycznych 59
Rys. 34 Schemat blokowy algorytmu poszukiwania punktów palca wzniesionego. 60
Rys. 35 Schemat blokowy algorytmu poszukiwania punktów palca zdjętego 61
Rys. 36 Fragment bazy obrazów. Każdy obraz posiada maskę regionu skóry oraz plik XML z zaznaczonymi punktami charakterystycznymi przez eksperta. 62
Rys. 37 Wnętrze pliku XML z opisem pozycji i typu punktów charakterystycznych dla Rys. 38. 63
Rys. 38 Wizualizacja oznaczonych punktów charakterystycznych, zapisanych w pliku XML z Rys. 37. 63
Rys. 39 Przykłady liczenia błędu dopasowania wykrytych punktów nadgarstka U', V' i W' do punktów wzorcowych U, V i W. Okręgi o promieniu E pokazują granicę znormalizowanego błędu. 66
Rys. 40 Graf prezentujący przykład rozwiązania problemu przedstawionego w Tab. 3 z użyciem algorytmu Dijkstry. Liniami przerywanymi oznaczono krawędzie, które poddano relaksacji. Optymalne rozwiązanie przedstawia kolor zielony. 68
Rys. 41 Przykład liczenia błędu dopasowania dla sparowanych punktów wykrytych (niebieskie) oraz punktów wzorcowych (zielone – nasady, żółte – czubki palców). Błąd dopasowania normalizowany jest przez szerokość nadgarstka wzorcowego (czerwony)..... 69
Rys. 42 Histogram błędu dopasowania punktu nadgarstka liczony dla 3 metod. 72
Rys. 43 Dystrybuanta błędu dopasowania punktu nadgarstka liczona dla 3 metod. 72
Rys. 44 Dystrybuanta błędu dla różnych interpretacji testów losowych..... 74
Rys. 45 Dystrybuanta błędu metod poszukiwania lokalnego minimum na profilu w celu wyznaczenia nadgarstka..... 74
Rys. 46 Dwa skrajne sposoby oznaczania obrazków przez ekspertów. Na obrazie a) oznaczono tylko punkty widoczne. Nasady palców i czubek jednego palca nie jest widoczny. Na obrazie b) zaznaczono punkty, nawet jeżeli nie były widoczne. Czubki palców oznaczono wewnątrz opuszka. 76
Rys. 47 Wykresy dystrybuanty błędów dla testów 5 algorytmów oraz błędu ludzkiego przeprowadzone na bazie hgr2a 77
Rys. 48 Wykresy dystrybuanty dla wyników 5 algorytmów działających na bazie hgr2a, przyrównanych do wskazań innego eksperta 79
Rys. 49 Wykresy dystrybuanty dla walidacji jedynie punktów czubków palców, testowanych na bazie hgr2a 80
Rys. 50 Wykresy dystrybuanty wyników testu łatwego 82

Rys. 51 Wykresy dystrybuanty wyników testu wpływu lokalizacji nadgarstka. Testy 1-3 przedstawiają wyniki algorytmu C2W. Testy 4-6 przedstawiają wyniki algorytmu autorskiego. Punkty nadgarstka wyznaczano metodami losowania najdłuższej przekątnej (random), wyznaczając momenty oraz pobierając punkty z bazy GT. 83
Rys. 52 Wykres kumulacji błędu w testach na bazie hgr1 84
Rys. 53 Wykres dystrybuanty wyników testu na bazie hgr1 85
Rys. 54 Wykresy dystrybuanty błędu testów na bazie hgr2b 86
Rys. 55 a) Obraz wzorcowy przedstawiający gest specjalny w kształcie litery „V", b-f) przykłady obrazujące działanie algorytmów, g) naniesienie wykrytych punktów oraz punktów wzorcowych przedstawionych szarymi kwadratami. 90
Rys. 56 a) Obraz wzorcowy przedstawiający gest „3", b-f) przykłady obrazujące działanie algorytmów, g) naniesienie wykrytych punktów oraz punktów wzorcowych przedstawionych szarymi kwadratami. 91
Rys. 57 a) Obraz wzorcowy przedstawiający gest „O", b-f) przykłady obrazujące działanie algorytmów, g) naniesienie wykrytych punktów oraz punktów wzorcowych przedstawionych szarymi kwadratami. 92
Rys. 58 a) Obraz wzorcowy przedstawiający gest „A", b-f) przykłady obrazujące działanie algorytmów, g) naniesienie wykrytych punktów oraz punktów wzorcowych przedstawionych szarymi kwadratami. 93
Rys. 59 Wpływ zmian parametru F_r na wykres kumulacji błędu i współczynnik jakości 96
Rys. 60 Wpływ zmian parametru t_{DT} na wykres kumulacji i współczynnik jakości 97
Rys. 61 Wpływ zmian parametru F_N na wykres kumulacji i współczynnik jakości 98
Rys. 62 Wpływ zmian parametru τ na wykres kumulacji i współczynnik jakości 99
Rys. 63 Wpływ zmian parametru P_N na wykres kumulacji i współczynnik jakości 100
Rys. 64 Wpływ zmian parametru λ na wykres kumulacji i współczynnik jakości 101
Rys. 65 Wpływ zmian parametru σ na wykres kumulacji i współczynnik jakości 102
Rys. 66 Wpływ zmian parametru t_E na wykres kumulacji i współczynnik jakości 103

Spis tabel

Tab. 1 Zbiór parametrów sterujących algorytmem SGONG wraz z ich opisem 30
Tab. 2 Koszty funkcji transformacji odległościowej 36
Tab. 3 Przykład macierzy wzajemnej odległości punktów wykrytych D do punktów wzorcowych P. Dobór par punktów zależny jest od wartości odległości pomiędzy punktami D i P, zwizualizowanych natężeniem szarości pól. Kolejne tabele przedstawiają różne metody parowania punktów. a) Wybór kolejnych wartości minimów globalnych, b) wybór tych pól, których suma jest najmniejsza, c) dopasowanie błędne, niezgodne z założeniami. 67
Tab. 4 Wyniki porównania testów algorytmów odnajdywania nadgarstka 73
Tab. 5 Skrócone nazwy algorytmów używane w trakcie testów 75
Tab. 6 Wyniki procedury walidacyjnej dla ustalenia błędu ludzkiego przeprowadzonego na bazie hgr2a. 78
Tab. 7 Wyniki procedury walidacyjnej algorytmów przeprowadzone na bazie hgr2a. 78
Tab. 8 Wyniki walidacji jedynie punktów czubków palców, testowanych na bazie hgr2a 81
Tab. 9 Wyniki walidacji testu łatwego 82
Tab. 10 Wyniki walidacji wpływu lokalizacji nadgarstka 83
Tab. 11 Wyniki walidacji testu na bazie hgr1 84
Tab. 12 Wyniki błędu testów na bazie hgr2a 86
Tab. 13 Zbiór parametrów sterujących algorytmem wyszukiwania punktów charakterystycznych dłoni 94
Tab. 14 Wpływ parametru L na czas i jakość poszukiwania nadgarstka w obrazach z bazy hgr2a. 95
Tab. 15 Wpływ zmian parametru F_r na wyniki testu 96
Tab. 16 Wpływ zmian parametru t_{DT} na wyniki testu 97
Tab. 17 Wpływ zmian parametru F_N na wyniki testu 98
Tab. 18 Wpływ zmian parametru τ na wyniki testu 99
Tab. 19 Wpływ zmian parametru P_N na wyniki testu 100
Tab. 20 Wpływ zmian parametru λ na wyniki testu 101
Tab. 21 Wpływ zmian parametru σ na wyniki testu 102
Tab. 22 Wpływ zmian parametru t_E na wyniki testu 103