POLITECHNIKA WARSZAWSKA

**Wydział Mechatroniki**

**Praca dyplomowa inżynierska**

Ireneusz Szulc

Inteligentny interfejs dotykowy umożliwiający obsługę złożonych gestów

Opiekun pracy:

mgr inż. Bogdan Harasymowicz-Boggio

Instytut Automatyki i Robotyki

Warszawa, 2016 r.

Bez numeru strony na stronie tytułowej

Oryginał karty pracy

Streszczenie w języku polskim (1 strona A4)

Streszczenie w języku angielskim (1 strona A4)

# Życiorys

# Spis treści

# 1. Wstęp

## 1.1. Cel pracy

Celem pracy jest zaprojektowanie i stworzenie inteligentnego interfejsu dotykowego umożliwiającego obsługę złożonych gestów. Taki interfejs w formie oprogramowania na urządzenie z ekranem dotykowym powinien być odpowiedzialny za rozpoznawanie sekwencji elementarnych konturów rysowanych przez użytkownika poprzez przeciąganie palca lub rysika po ekranie. Wynik analizy wprowadzonych kombinacji gestów może posłużyć do rozpoznawania znaków odręcznego pisma lub wykonania zdefiniowanych akcji w systemie, na którym pracuje urządzenie. Stworzony system pozwala użytkownikowi na swobodne definiowanie własnych gestów o dowolnej złożoności.

Inteligencja powstałego interfejsu polega na zdolności systemu do samouczenia się, co w praktyce oznacza, że system jest w stanie zwiększać swoją skuteczność w rozpoznawaniu znaków poprzez dostosowywanie się do charakteru pisma odręcznego użytkownika oraz zmianę swojego zachowania wskutek akcji wykonywanych przez użytkownika.

## 1.2. Założenia projektu

Założeniem projektu jest, aby interfejs dotykowy miał formę aplikacji działającej na urządzeniu mobilnym wyposażonym w system operacyjny Android. Dodatkowo takie urządzenie powinno posiadać ekran dotykowy, który posłuży za źródło danych wejściowych dla aplikacji. Językiem programowania wykorzystanym do stworzenia całej aplikacji jest język Java.

Interfejs powinien umożliwiać rozpoznawanie znaków pisma odręcznego wprowadzanych przez użytkownika poprzez ruch palca lub rysika po ekranie dotykowym, zamianę ich na tekst w formie cyfrowej i przechowywanie go w celu dalszego wykorzystania. Wprowadzane znaki mogą składać się zarówno z wielu fragmentów pisanych z odrywaniem rysika od ekranu, jak również z pojedynczych konturów.

Dodatkowym założeniem jest, aby system uczył się w trakcie działania. Aplikacja powinna nauczyć się charakteru pisma odręcznego od użytkownika, powinna także zwiększać swoją skuteczność w rozpoznawaniu gestów w miarę dłuższego czasu użytkowania. Implikuje to konieczność przechowywania dodatkowych danych zbieranych w trakcie działania aplikacji w celu dostosowywania się do charakteru pisma użytkownika. Cały system powinien również reagować na polecenia uczące wydawane przez użytkownika i zmieniać sposób swojego działania.

Zakres pracy obejmuje trzy główne zagadnienia:

* Projekt i implementację algorytmu rozpoznawania elementarnych gestów,
* Opracowanie metody rozpoznawania złożonych gestów oraz wprowadzania znaków na podstawie pisma odręcznego,
* Projekt i implementacja oprogramowania na urządzenie mobilne.

Praca skupia się głównie na rozpoznawaniu znaków pisma odręcznego, ale łatwo można rozszerzyć tą funkcjonalność systemu na rozpoznawanie gestów wykonujących określone akcje w systemie (np. skrótowe uruchamianie aplikacji), gdyż zaprojektowany system nie ma ograniczenia rozpoznawania dowolnych gestów o dowolnej złożoności.

## 1.3. Zastosowanie

Opracowany system może znaleźć zastosowanie jako wygodna metoda wprowadzania znaków na urządzeniach z panelem dotykowym. Takie rozwiązanie nie ogranicza się jedynie do wprowadzania tekstu, ale pozwala także na równie szybkie wprowadzanie cyfr, znaków diakrytycznych (polskich liter) oraz znaków specjalnych (takich jak: przecinek, kropka, wykrzyknik, itd.), co może przemawiać na korzyść metody w porównaniu do konwencjonalnych klawiatur ekranowych, gdzie wprowadzanie znaków specjalnych może stać się uciążliwe i czasochłonne. Metoda ta jest bardzo intuicyjna, charakteryzuje się prostotą i mogłaby być chętniej wykorzystywana np. przez ludzi starszych przyzwyczajonych do tradycyjnego wprowadzania tekstu.

Kolejnym polem zastosowań są urządzenia z bardzo małymi ekranami dotykowymi (np. smartwatche), gdzie często wykorzystanie klawiatury ekranowej może być uciążliwe z powodu bardzo małych przycisków, zaś wprowadzenie sterowania urządzeniem poprzez gesty oraz pisania dowolnych znaków metodą pisma odręcznego mogłoby zwiększyć komfort użytkowania takiego urządzenia.

System może posłużyć również jako metoda wprowadzania haseł w postaci sekwencji gestów w celu uzyskania dostępu do określnych zasobów. Przetrzymywanie hasła w postaci wzorca charakteru pisma może być bardzo pomocne w procesie uwierzytelniania użytkowników i przyznawać dostęp jedynie osobie, która ma dokładnie taki sam charakter pisma lub w dokładnie taki sam sposób wykonała określone gesty. System z takim rozwiązaniem nie przyznawałby dostępu osobie, która poznała hasło, lecz wprowadziła je w inny sposób.

Interfejs dotykowy będący przedmiotem pracy można również wykorzystać jako rozszerzenie systemu, na którym pracuje urządzenie do skrótowego uruchamiania aplikacji lub niektórych funkcji systemu. Opracowaną metodę można zaimplementować także na panelach operatorskich HMI z ekranem dotykowym w celu wygodnego i łatwego przeglądania monitorowanych wielkości i wizualizowanych procesów w przemyśle.

## 1.4. Przegląd istniejących rozwiązań

gotowe rozwiązania na rynku, opis, oprogramowanie w smatrfonach

przegląd metod on-line Handwriting Recognition, opis

inne prace naukowe

# 2. Wstęp teoretyczny



"...syntaktyczne rozpoznawanie obrazów - reprezentowanie obiektu przez ciąg...

ten obrazek, opis

segmentacja, klasyfikacja, przetwarzanie wstępne

on-line Handwriting Recognition

off-line Handwriting Recognition

## 2.1. Klasyfikatory

### 2.1.1. Kody łańcuchowe Freemana



łańcuchy Freemana - zalety: mała zajętość pamięci, prostota

Histogramy - co to?

### 2.1.2. Sieci neuronowe

### 2.1.3. Supported Vector Machine

inne prace, źródła, jak inni to robili i co stwierdzili

## 2.2. Metody minimalnoodległościowe

Metoda najbliższych sąsiadów

duża wrażliwość na błędy ciągu uczącego

Crosss-Correlation?, [Tadeusiewicz]

## 2.3. Metody korelacji

### 2.3.1. Korelacja Pearsona

wzory matematyczne

### 2.3.2. Inne korelacje

Batacharraya, itd.

## 2.3. Słownik używanych pojęć

Stosowane pojęcia:

**Gest elementarny, gest składowy, gest pojedynczy** - kontur powstały w wyniku przesunięcia palca lub rysika po ekranie bez odrywania go od ekranu

**Gest złożony** - sekwencja następujących po sobie gestów elementarnych, wieloelementowy gest

**Złożoność gestu -** liczba gestów elementarnych składających się na gest złożony

punkt startowy

zbiorczy współczynnik korelacji - wypadkowy

zkorygowany współczynnik korelacji

## 2.4. Info o systemie Android

łatwość prorgramowania

## 2.5. Info o OpenCV

OpenCV ma implementację na system Android w Javie, licencja darmowa

# 3. Algorytm rozpoznawania pojedynczych konturów

Rozdział ten skupia się na opracowaniu i zaimplementowaniu metody rozpoznawania pojedynczych gestów składowych (bez rozważania kontekstu, w jakim zostały wprowadzone).

Ogólny proces rozpoznawania obiektów można podzielić na kilka zasadniczych części:

* przetwarzanie wstępne danych wejściowych,
* segmentacja (wyodrębnienie cech charakterystycznych obiektu),
* utworzenie reprezentacji nieznanego obiektu,
* właściwe rozpoznawanie (klasyfikacja).

Zastosowany algorytm opiera się na jednej z metod minimalnoodległościowych - metodzie najbliższych sąsiadów. Polega ona na bardzo prostym rozumowaniu mówiącym, że wynikiem rozpoznania jest klasa, do której należy obiekt najbliższy (w myśl przyjętej metryki) rozpoznawanemu obiektowi.

W przypadku rozważanego w projekcie zagadnienia rozpoznawania konturów (będących danymi wejściowymi dla algorytmu), obiektami są właśnie poszczególne kontury, które mogą być pogrupowane w klasy. Obiekty te leżą w wielowymiarowej przestrzeni (liczba wymiarów przestrzeni jest równa liczbie cech obiektów). Położenie obiektów jest więc w pełni uwarunkowane cechami obiektów, natomiast metryka określa w tym przypadku stopień podobieństwa konturów (jak bardzo dwa obiekty są do siebie zbliżone). Na tej podstawie można wyznaczyć stopień podobieństwa nowych danych wejściowych (niesklasyfikowanych) do każdego obiektu, który już został sklasyfikowany. Następnie można przyporządkować nieznanemu obiektowi klasę, do której należy najbardziej zbliżony (o minimalnej odległości) znany obiekt. Zasadniczym problemem w tej metodzie staje się zdefiniowanie samej metryki określającej odległość między obiektami (stopień ich podobieństwa). Taka metoda implikuje również konieczność przechowywania wzorców w wewnętrznej bazie aplikacji, które przynależą już do określonych klas i mogą stać się podstawą do identyfikacji nowo wprowadzonych, nieznanych obiektów.

## 3.1. Założenia algorytmu

Aby uzyskać jak największą skuteczność w poprawnym rozpoznawaniu elementarnych gestów i jednocześnie zachować pewną tolerancję danych wejściowych (nowych, nierozpoznanych konturów) na zniekształcenia w stosunku do wzorca, konieczne jest rozważenie problematycznych przypadków oraz określenie wymagań w stosunku do algorytmu. Może to pomóc w uniknięciu błędnego działania algorytmu w przyszłości.

Każdy gest pisany przez użytkownika jest zniekształconym naśladownictwem pewnego abstrakcyjnego idealnego wzorca, do którego dążą w sposób niedoskonały wszystkie konkretne realizacje obiektów z danej klasy. Należy więc pamiętać o tym, że proces identyfikacji konturów powinien być niewrażliwy (jedynie w odpowiednim zakresie) na wybrane transformacje obrazu.

Rozważany algorytm powinien spełniać poniższe założenia wynikające z charakteru wykonywania gestów przez użytkownika:

* uwzględnianie kierunku wykonywania gestu - Program powinien rozróżniać między sobą gesty, które mają ten sam kształt konturu, lecz powstały w wyniku odwróconej sekwencji ruchów na ekranie dotykowym. Właściwość kierunku konturu można odczytać tylko w przypadku metod on-line Handwriting Recognition, gdyż znana jest cała historia i dynamika powstawania konturu. Metoda ta śledzi na bieżąco cały proces powstawania konturu. Natomiast w przypadku metod off-line Handwriting Recognition, nie można stwierdzić, jaki był kierunek powstawania konturu na podstawie samego obrazu (np. z kamery), będącego jedynie końcowym efektem pisania znaku. Znajomość cechy kierunku jest dużą zaletą metody, gdyż wykorzystanie jej może znacznie poprawić skuteczność rozpoznawania algorytmu.
* częściowa niewrażliwość na obrót (jedynie w odpowiednim zakresie) - Użytkownik powinien mieć możliwość wpisania znaku w nieco innej, zniekształconej orientacji, ale tylko w ustalonym, niewielkim zakresie. Pełna niewrażliwość na obrót może prowadzić do błędnego, jednakowego interpretowania niektórych znaków, np. dużych liter "M" i "W", "N" i "Z", lub cyfr "6" i "9", których ogólny kształt może różnić się jedynie orientacją, co mogłoby prowadzić do nierozróżniania takich znaków między sobą.
* częściowa niewrażliwość na przesunięcie (jedynie w odpowiednim zakresie) - Za każdym razem gest wykonywany jest nieco inaczej przez użytkownika, również punkt rozpoczęcia rysowania znajduje się w różnych miejscach. Tutaj również pełna niewrażliwość na przesunięcie konturu w układzie współrzędnych ekranu nie sprawdziłaby się w pewnych przypadkach, np. napisanie znaku "ż" mogłoby być interpretowane przez aplikację jako kombinacja litery "z" i kropki ".". W takiej sytuacji znajomość umiejscowienia kropki nad literą lub w dolnej części ekranu może rozsądzić problematyczną analizę i dać poprawny rezultat. Podobna sytuacja mogłaby wystąpić w przypadku litery "ć" oraz kombinacji litery "c" i przecinka ",".
* częściowa niewrażliwość na skalowanie (jedynie w odpowiednim zakresie) - podobnie jak w poprzednich przypadkach właściwość ta powinna obowiązywać tylko w ustalonym zakresie. Ma to na celu lepsze rozpoznawanie między sobą małych i dużych liter alfabetu, które w niektórych przypadkach charakteru pisma mogą różnić się jedynie wielkością.
* wydobycie ogólnych cech obiektów - Algorytm nie powinien analizować szczegółowych kształtów konturu a powinien w miarę możliwości aproksymować jego cechy geometryczne.

## 3.2. Wybór metody

Do identyfikacji obiektów będących konturami zdecydowano się wykorzystać kody łańcuchowe Freemana oraz reprezentację cech geometrycznych konturu w postaci histogramu obrazującego rozkład tych kodów. Dodatkowo informację o obiekcie rozszerza również informacja o współrzędnych punktu na ekranie, w którym gest zaczął być rysowany oraz liczba pikseli, z których składa się kontur wyrażająca jego długość. Podsumowując, pełna informacja o cechach geometrycznych pojedynczego gestu składowego, która wystarcza do identyfikacji obiektów, reprezentowana jest i przechowywana w trzech elementach:

* histogram kodów łańcuchowych Freemana,
* współrzędne punktu początku konturu,
* całkowita liczba pikseli konturu.

Przewiduje się, że cała aplikacja z kompletną bazą wzorców będzie przechowywać nawet do kilkuset wzorców po długim czasie użytkowania przez użytkownika. Dlatego też od aplikacji oczekuje się, że będzie rozpoznawać znaki w możliwie najkrótszym czasie tak, aby dać użytkownikowi wrażenie płynnego wprowadzania tekstu. Z tego względu kody łańcuchowe Freemana jako metoda reprezentacji obiektu wydają się być dobrym wyborem z perspektywy małej zajętości w pamięci oraz stosunkowo szybkiego algorytmu porównywania wzorców, gdyż dane histogramu charakteryzujące kształt przechowywane mogą być jako tablica jedynie ośmiu liczb zmiennoprzecinkowych. Dokładny opis algorytmu i zasada działania zostały przedstawione w kolejnych rozdziałach.

opis, jakie metody inni wybrali w innych pracach

## 3.3. Przetwarzanie wstępne

Danymi wejściowymi dla algorytmu rozpoznawania elementarnych gestów jest sekwencja punktów, z jakich składa się kontur. Akwizycja tych danych odbywa się dzięki interakcji z systemem, na jakim pracuje urządzenie. Dane o położeniu palca lub rysika na ekranie dotykowym w systemie Android możliwe są do odczytania tylko co pewien, krótki okres czasu. Oznacza to, że gest wykonany przez użytkownika w sposób szybki może być odczytany z mniejszą dokładnością. Każdy gest poddawany jest aproksymacji (w mniejszym lub większym stopniu), a więc z ekranu dotykowego można odczytać zbiór punktów, który reprezentuje linię łamaną otwartą. Po sygnale oderwania palca od ekranu można łatwo stwierdzić, że rysowanie konturu zostało zakończone i można poddać go dalszej analizie.

Zaraz po odczycie, gest składowy poddawany jest przetwarzaniu wstępnemu. Pierwsza część polega na uzupełnieniu listy punktów o brakujące piksele tak, aby lista zawierała tyle pikseli, z ilu faktycznie składa się kontur na ekranie. Liczba uzupełnianych pikseli proporcjonalna jest do odległości między punktami w metryce euklidesowej dla każdej pary sąsiednich punktów. Przekształcenie to konieczne jest to do poprawnej dalszej analizy gestu. Do wyznaczenia współrzędnych punktów pomiędzy sąsiednimi pikselami wykorzystywana jest interpolacja liniowa współrzędnych punktów będących najbliższymi sąsiadami. Należy zaznaczyć, że współrzędne interpolowanych pikseli zapisywane są jako liczby zmiennoprzecinkowe a nie jako całkowite, aby zapisywać ich położenie z większą dokładnością.

Kolejnym etapem jest filtracja konturu otrzymanego z poprzedniego kroku. Polega to na wyznaczeniu nowego konturu, który powstaje poprzez uśrednienie pewnej liczby sąsiednich punktów i utworzenie nowych współrzędnych punktu. Filtracja ma na celu wygładzenie konturu oraz redukcję szumów (drgań rysika podczas rysowania oraz kompensacji błędów niedokładności czujnika dotykowego). Przeprowadzone doświadczenia pokazały, że zwiększa to skuteczność w późniejszym wyznaczaniu wektorów kierunków między pikselami i wpływa na wyraźniejszy rozkład łańcuchów Freemana na histogramie, zmniejszając ilość wystąpień błędnie rozpoznanych kierunków, które wynikają z porównywania jedynie dwóch sąsiednich punktów znajdujących się bardzo blisko siebie. Liczba sąsiadujących pikseli, z jakich wyznaczany jest nowy uśredniany punkt wyznaczona została eksperymentalnie i może zależeć od rozdzielczości ekranu oraz od gęstości pikseli na ekranie (liczby DPI).

Poniżej przedstawiono ideę przetwarzania wstępnego na przykładzie prostego konturu oraz efekt poszczególnych etapów. Kropki na rysunku reprezentują punkty przechowywane w liście.



Rysunek ?????. Przykładowy kontur poddany przetwarzaniu wstępnemu: a) punkty odczytane ze sterownika ekranu dotykowego przed przetwarzaniem wstępnym, b) punkty uzyskane w wyniku uzupełnienia i interpolacji, c) lista uśrednionych punktów po etapie filtracji

Dzięki takiemu podejściu z konturu wydobywane są jego ogólne cechy i przybliżony kształt, zaś szczegółowe cechy geometryczne mogą zostać pominięte na skutek filtracji.

## 3.4. Segmentacja

Do segmentacji wykorzystano kody łańcuchowe Freemana. W skrócie polega to na tym, że dla każdej pary sąsiednich pikseli z konturu obliczana jest różnica współrzędnych i na tej podstawie wyznaczany jest kąt, jaki tworzy powstały odcinek z niezmienną osią poziomą ekranu (za pomocą funkcji ). Następnie wyznaczany jest jeden z 8 kierunków, do których przynależy taki kąt. Szerokość jednego przedziału kątów dla każdego kierunku jest taka sama i jest równa ilorazowi kąta pełnego i liczby wszystkich przedziałów kierunków.



Rysunek ?????. Podział kąta pełnego na 8 przedziałów kierunków

Do zapisania całego konturu w postaci kodów łańcuchowych Freemana potrzebna jest lista liczb całkowitych, z których każda reprezentuje kierunek zmiany współrzędnej punktu w stosunku do poprzedniego piksela i wyraża się liczbą z zakresu od 0 do 7.

Taki sposób reprezentacji ma pożądaną cechę - częściową niewrażliwość na obrót. W niektórych sytuacjach obrót całego wejściowego konturu nawet do 45 stopni może skutkować zapisaniem dokładnie tymi samymi wartościami kodów łańcuchowych (np. w przypadku kształtu linii prostych). Warto zauważyć również, że taki zapis w postaci kodów łańcuchowych jest silnie związany z kierunkiem wykonywania gestu, co jest istotną zaletą, gdyż dzięki temu można łatwo odróżnić wiele gestów, np. pionową linię pisaną w dół lub górę.

## 3.5. Reprezentacja obiektu

Kolejnym etap jest wyznaczenie z listy kodów łańcuchowych Freemana histogramu liczby ich występowania. Polega to oczywiście na zliczeniu ilości wystąpień dla każdego takiego kodu, a następnie podzieleniu tej liczby przez ilość wszystkich kodów występujących w konturze w celu normalizacji histogramu.

Taka reprezentacja obiektu jest bardzo wygodna w dalszej analizie (porównywaniu histogramów). Warto również zauważyć, że takie przedstawienie obiektu ma właściwość całkowitej niewrażliwości na przesunięcia (histogramy kodów łańcuchowych będą dokładnie takie same w wyniku dowolnego przesunięcia konturu) oraz całkowitej niewrażliwości na skalowanie wejściowego konturu (z powodu normalizacji histogramu rozkład procentowy każdego z kierunków będzie również taki sam).

Ze względu na założenie częściowej niewrażliwości na przesunięcie oraz skalowanie w procesie rozpoznawania gestów elementarnych konieczne jest przechowywanie dodatkowych informacji o obiekcie, które mogą być wykorzystane podczas wyznaczania podobieństwa obiektów, a które ograniczą w pewnym stopniu całkowitą niewrażliwość konturu na wybrane przekształcenia geometryczne. Do tego celu razem z histogramem kodów łańcuchowych wyrażającym sam kształt gestu elementarnego dołączana jest również informacja o współrzędnych punktu startowego dla konturu oraz jego długości (wyrażającym wielkość gestu).

## 3.6. Klasyfikacja

Aby dokonać identyfikacji nieznanego gestu elementarnego, należy wyznaczyć współczynnik stwierdzający stopień podobieństwa gestów. W przypadku, gdy istnieje wiele cech obiektów, rozwiązaniem może być wyznaczenie współczynników podobieństwa dla każdej cechy, a następnie obliczenie zbiorczego współczynnika korelacji.

### 3.6.1. Korelacja histogramów

Algorytm identyfikacji pojedynczych gestów wymaga metody porównywania wprowadzonych konturów z konturami przechowywanymi w bazie aplikacji jako wzorce w celu wyznaczenia współczynnika podobieństwa każdego do z nich i wybrania najlepszego dopasowania. Opisywany współczynnik to współczynnik korelacji liniowej Pearsona. Mieści się w zakresie i większa jego wartość oznacza większy stopień podobieństwa porównywanych ciągów liczb. Danymi wejściowymi dla korelacji Pearsona są histogramy kodów łańcuchowych określających kształt gestów. Dzięki temu wynikiem takiego zestawienia histogramów jest liczba określająca jak bardzo kształty gestów są do siebie zbliżone.

Wyznaczenie współczynnika korelacji Pearsona odbywa się poprzez wykorzystanie funkcji z biblioteki OpenCV. Obliczanie trwa bardzo szybko z powodu między innymi bardzo małej ilości danych do przeanalizowania (porównanie 2 tablic o rozmiarze 8) oraz zoptymalizowanego algorytmu zaimplementowanego w bibliotece OpenCV.

### 3.6.2. Korelacja punktów startowych

Aby stwierdzić podobieństwo dotyczące punktów startowych dla konturów, obliczana jest odległość między punktami, następnie dzielona jest przez mniejszy wymiar ekranu (liczbę pikseli szerokości lub wysokości ekranu) tak, aby uzyskać wielkość niezależną od wielkości ekranu urządzenia, na jakim uruchomiona jest aplikacja. Tak uzyskana liczba nazwana jest względną odległością punktów. Wyznaczenie współczynnika korelacji punktów startowych zależne jest również od 2 parametrów wybranych doświadczalnie. Są to współczynniki oraz . Wartość współczynnika określa jaki jest próg tolerancji dla przesunięcia punktów startowych między sobą i nie wpływa ono jeszcze na współczynnik korelacji punktów startowych (uznaje, że rozpoczęcie rysowania gestów odbyło się w tym samym miejscu). Natomiast współczynnik określa jak duża powinna być różnica w odległości punktów między sobą, aby uznać, że nie występuje jakiekolwiek podobieństwo, jeśli chodzi o miejsce rozpoczęcia rysowania. W zakresie względnej odległości punktów między a wartość współczynnika korelacji interpolowana jest w sposób liniowy, co przedstawia się na poniższym wykresie.



Rysunek ?????. Wykres zależności współczynnika korelacji punktów startowych od wartości względnej odległości punktów

Wartości współczynników i zostały wybrane na drodze doświadczalnej i wynoszą odpowiednio: oraz .

### 3.6.3. Korelacja długości gestu

Do wyznaczenia podobieństwa gestów dotyczącego ich wielkości posłużono się współczynnikiem korelacji długości. Do tego celu należy wyznaczyć współczynnik będący względną różnicą długości gestów, wyraża się on wzorem:

gdzie - większa wartość długości konturu spośród porównywanych konturów,   
 - mniejsza wartość długości konturu spośród porównywanych konturów.

Podobnie jak w poprzednim przypadku, współczynnik korelacji długości zależny jest od 2 parametrów wybranych doświadczalnie. Są to współczynniki oraz . Wartość współczynnika określa martwą strefę niewrażliwości na skalowanie (dla algorytm uznaje, że gesty są takiej samej wielkości). Zaś współczynnik określa górną granicę wrażliwości na skalowanie (dla składowe gesty nie wykazują żadnego podobieństwa, jeśli chodzi o wspólną wielkość. W zakresie względnej różnicy długości między a wartość współczynnika korelacji interpolowana jest w sposób liniowy, przebieg jego funkcji przedstawiono na poniższym wykresie:



Rysunek ?????. Wykres zależności współczynnika korelacji długości od wartości względnej różnicy długości

Wartości współczynników i zostały wybrane na drodze doświadczalnej i wynoszą odpowiednio: oraz .

### 3.6.4. Zbiorczy współczynnik korelacji

Aby dokonać identyfikacji nieznanego gestu elementarnego, należy wyznaczyć współczynnik stwierdzający stopień podobieństwa gestów. W przypadku, gdy istnieje wiele cech obiektów, rozwiązaniem może być wyznaczenie współczynników podobieństwa dla każdej cechy, a następnie obliczenie zbiorczego współczynnika korelacji

W celu rozpoznania wprowadzonego gestu za wynik uznaje się ten wzorzec, dla którego współczynnik korelacji był największy (i odpowiednio duży).

Można stwierdzić, że elementarne gesty należą do tej samej klasy gestów, gdy ich współczynnik korelacji jest większy lub równy pewnemu ustalonemu progowi

obliczanie korelacji wypadkowej za pomocą średniej ważonej, dobór współczynników

## 3.5. Osobny przypadek krótkich konturów - Kropki

konieczność osobnego traktowania kropek

z powodów: za mało danych do analizy (filtracja sobie nie radzi) + za duże szumy, byłaby duża rozbieżność

## 3.7. Dokładny opis algorytmu

filtracja szumów

dokładny opis algorytmu rozpoznawania pojedynczych gestów lub wyznaczania samej korelacji,

pamiętanie histogramu, punktu startu i długości gestu składowego

schematy blokowe,

przykładowe histogramy pojedynczych gestów

## 3.8. Przykłady

przykłady histogramów charakterystycznych liter,

jak wygląda histogram bez filtracji

# 4. Projekt modułu wprowadzania znaków na podstawie pisma odręcznego

## 4.1. Rozpoznawanie złożonych gestów

rozpoznawanie ciągu pojedynczych gestów - złożonych gestów - algorytm

korelacja złożonego gestu jako średnia arytmetyczna gestów składowych,

kryterium wyboru najlepszego rozwiązania - decyduje sumaryczny współczynnik korelacji

schematy blokowe do skomplikowanych algorytmów

wpływ złożoności gestu na zkorygowany współczynnik korelacji i znajdowanie najlepszego dopasowania

rozszerzenie o niektóre gesty: np. spacja, enter, backspace

## 4.2. Inteligencja systemu

poważny problem - błędne wzorce, prowadzące do błędnej klasyfikacji

inteligencja, uczenie w trakcie działania

usuwanie wzorców, które prowadzą do błędnego rozpoznawania

uczenie: dodawanie wszystkich wzorców, usuwanie wzorców, które są rzadko podstawą do rozpoznania, optymalizator wzorców,

pytanie w przypadku nie rozpoznania żadnego wzorca, co to było?

zapisywanie liczby gestów dobrze rozpoznanych przez wzorzec i źle rozpoznanych

przycisk poprawiania gestu: zanotowanie błędnego rozpoznania i zastąpienie gestu

usuwanie nadmiarowych obiektów ciągu uczącego

## 4.1. Automatyczne dodawanie wzorców

zapisywanie dodatkowej informacji o gestach dodanych automatycznie

automatyczne dodawanie gestów o słabym współczynniku korelacji

w przypadku poprawiania lub usuwania znaku, usuwanie gestu automatycznie dodanego i zapisanie złego rozpoznania przez gest, który go dodał

## 4.2. Optymalizator wzorców

obcinanie i zapis listy wzorców przy wyjściu z programu

obcinanie wzorców dla znaków, które mają za dużo wzorców, wybieranie najlepszych (lub najnowszych)

usuwanie wzorców z powodu zbyt niskiego bilansu rozpoznania

mechanizm usuwania złych wzorców mających dużą liczbę błednych rozpoznań

# 5. Implementacja oprogramowania na urządzenie mobilne

## 5.1. Urządzenie

opis urządzenia, ekran dotykowy, rozdzielczość, parametry procesora (czas trwania analizy, obliczeń)

nie musi być mobilne, wystarczy ekran dotykowy (panele HMI)

## 5.2. Struktura aplikacji

diagram klas aplikacji, struktura aplikacji, wzorce projektowe

## 5.3. Działanie aplikacji

opis działania, przechodzenia między ekranami

fragmenty niektórych kodów?, funkcje ?

OpenCV - użyte funkcje

Android - funkcje

screeny z programu

system gotowy do nauki i użytkowania już w przypadku pustej bazy wzorców

zapis do pliku - mechanizm serializacji obiektów w Javie

tryb menedżera gestów i szybkiego pisania, opis przycisków, funkcji

# 6. Testy aplikacji

## 6.1. Wybór metod i dobór współczynników

eksperymentalny dobór współczynników

testy i wyniki różnych metod korelacji w OpenCV (wzorki, wykresiki, tabelki), eksperymenty z normalizacją histogramów i współczynnikami, dobór najlepszej metody

testy aplikacji, rozpoznawania tekstu, procent poprawnych rozpoznań, czas działania

wartości współczynników z Configa

## 6.2. Testy końcowe działającej aplikacji

## 6.3. Statystyki

statystyki: liczby gestów, wzorców, procent dobrych rozpoznań

# 7. Bibliografia

Wykaz literatury

Załączniki

Plakat

na plakacie (posterze) dyplomu powinny znaleźć się m.in. następujące informacje

1. kierunek studiów

2. specjalność

3. rok akademicki ukończenia studiów