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 wykonywanych przez użytkownika poprzez przeciąganie palca lub rysika po ekranie. Wynik analizy wprowadzonych gestów może posłużyć do rozpoznawania znaków odręcznego pisma lub wykonania zdefiniowanych akcji w systemie, na którym pracuje urządzenie.

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.

Praca wchodzi w zakres widzenia maszynowego

Złożone gesty: znaki - odręczne pismo, obsługa akcji dla dowolnych gestów dowolnej długości

Nie tylko znaki, obsługa gestów w celu np. skrótowego uruchamiania aplikacji

skupienie się na piśmie odręcznym, ale łątwo można przenieść na rozpoznawanie innych gestów

samouczący się system rozpoznawania gestów

## 1.2. Założenia i wymagania 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 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 3 punkty - opis zakresu

## 1.3. Zastosowanie

Opracowany system może znaleźć zastosowanie jako wygodna metoda wprowadzania znaków na urządzeniach z panelem dotykowym. Taka metoda 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 dać jej przewagę nad konwencjonalnymi klawiaturami ekranowymi, gdzie wprowadzanie znaków specjalnych często staje się uciążliwe i wolniejsze. Metoda ta jest bardzo intuicyjna i charakteryzuje się prostotą, i mogłaby być chętniej wykorzystywana 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 bardzo uciążliwe z powody małych drobnych przycisków, zaś wprowadzenie sterowania urządzeniem poprzez gesty oraz wprowadzania dowolnych znaków metodą pisma odręcznego może zwiększyć znacznie komfort użytkowania takiego urządzenia.

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

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. W ten sposób system nie przyznawałby dostępu osobie, która poznała hasło, lecz wprowadziła je w inny sposób.

## 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

**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

korelacja wypadkowa

## 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 w przypadku metod off-line Handwriting Recognition metody te nie są w stanie stwierdzić, jaki był kierunek powstawania konturu na podstawie samego obrazu, będącego jedynie końcowym efektem pisania znaku. Jest to duża zaleta i dobre wykorzystanie tej cechy może znacznie poprawić skuteczność rozpoznawania algorytmu. Metoda on-line śledzi na bieżąca cały proces powstawania konturu.
* niewrażliwość na obrót (jedynie w odpowiednim zakresie) - powinno się umożliwić użytkownikowi wpisanie 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 lub N i Z, których ogólny kształt różni się jedynie orientacją i może prowadzić do nie rozróżniania takich znaków między sobą.
* powinien wydobywać ogólne cechy obiektów, nie analizować szczegółów

algorytm powinien być również odporny na obrót ale tylko w pewnym zakresie, gdyż odporność na obrót może prowadzić np. do takiego samego postrzegania dużych liter W oraz M, mających wspólny środek symetrii. (lub liter N i Z)

algorytm powinien być odporny na przesunięcia znaków (gdyż za każdym razem znak jest wpisywany nieco inaczej). odporność na przesunięcie tylko w pewnym zakresie, np. Ć i C i przecinek

odporność na skalowanie tylko w pewnym zakresie, uwzględnienie długości

uwzględnianie kierunku pisania

uwzględnienie początku rysowania

## 3.2. Wybór metody

Od aplikacji oczekuje się, że będzie rozpoznawać znaki w jak najmniejszym czasie, aby dać użytkownikowi możliwość płynnego wprowadzania tekstu.

Z tego względu zdecydowano się na reprezentację wprowadzanych konturów za pomocą łańcuchów Freemana.

kody łańcuchowe Freemana - znacznie mniej pamięciochłonne

Ze względu na szybkie działanie.

wybór metody, opis, dlaczego

Wymagania metody:

odporność na obrót tylko w pewnym zakresie

odporność na przesunięcie tylko w pewnym zakresie

odporność na skalowanie tylko w pewnym zakresie

uwzględnianie kierunku pisania

uwzględnienie początku rysowania

## 3.3. Przetwarzanie wstępne

Filtracja

Danymi wejściowymi są kolejne punkty z ekranu dotykowego wysyłane okresowo co pewien czas.

Konieczna jest zamiana tych punktów na listę kolejnych pikseli (zastosowanie interpolacji liniowej punktów)

Przetwarzanie wstępne ogranicza się jedynie do filtracji drgań palca, niedokładności czujnika dotykowego położenia rysika - filtracja szumów, uśrednianie

## 3.4. Segmentacja

## 3.5. Reprezentacja obiektu

## 3.6. Klasyfikacja, Korelacja

Wynikiem każdego takiego porównania jest współczynnik korelacji określający jak bardzo gesty są do siebie zbliżone. W celu rozpoznania wprowadzonego gestu za wynik uznaje się ten wzorzec, dla którego współczynnik korelacji był największy (i odpowiednio duży).

korelacja: histogramu, punktu startowego i długości gestu

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,

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

## 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

## 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