

**Akademia Górniczo - Hutnicza
im. Stanisława Staszica w Krakowie**



AGH

Projekt

z przedmiotu

Analiza i Przetwarzanie Obrazów

Wykrywanie tablic rejestracyjnych

Jagięła Michał
IS (GKiPO)

1. Cel oraz założenia projektu

Celem projektu było stworzenie programu automatycznie wykrywającego tablice rejestracyjne na zdjęciach. Jest to pierwszy i najtrudniejszy etap rozpoznawania tablic rejestracyjnych.

Program miał być napisana w języku Java i środowisku Fiji. Z powodu problemów z aplikacją Fiji, z którymi napotkałem się na laboratoriach aplikacja została stworzona w środowisku NetBeans, w podobny sposób jak tworzone były programy na laboratoriach, ale może być uruchamiany za pomocą Fiji.

Można rozróżnić dwie kategorie systemów identyfikacji tablic rejestracyjnych:

- rozpoznawanie statyczne (dla pojazdów nieruchomych)
- rozpoznawanie dynamiczne (dla pojazdów ruchomych)

W poniższym projekcie założono rozpoznawanie statyczne. Założenia systemu: Pojazd podlegający rozpoznaniu, nie może się poruszać, a ponadto powinien znajdować się w określonym położeniu względem kamery. Często wymuszane jest to poprzez instalowanie szlabanów oraz bramek wjazdowych. Tego typu system posiada możliwość rozpoznawania jednocześnie tylko jednego pojazdu, jednak dzięki temu dokładność rozpoznawania jest wyższa niż w systemach wykorzystujących rozpoznawanie dynamiczne. Pozwala to stosować systemy rozpoznawania statycznego np. na parkingach, gdzie jakość rozpoznawania jest szczególnie ważna.

2. Opis programu i algorytmu

Program został napisany w języku Java w środowisku NetBeans, jako wtyczka aplikacji Fiji.

Program został stworzony na zasadzie algorytmu wykrywania krawędzi i rzutu jasności. Zakłada on, że tablice rejestracyjne występują zazwyczaj na obrazie jako fragmenty o bardzo wysokim kontraście, czarne litery na białym, ewentualnie żółtym, tle (nowe tablice), lub białe litery na czarnym tle. Dodatkowo zakłada, że znaki tablicy znajdują się w jednym rzędzie, co powoduje dużą częstotliwość zmian jasności obrazu w linii poziomej. Pozwala to na szukanie tablic rejestracyjnych w wierszach w których występuje duża liczba wyraźnych zmian jasności. Zgodnie z tym założeniem algorytm wyznacza najpierw liczbę zmian jasności w każdym wierszu, uśrednia je, po czym wyszukuje wierszy z największą liczbą zmian jako prawdopodobna lokalizacja tablicy. Podobny proces wykonywany jest następnie dla kolumn, dla wyznaczonego wcześniej prawdopodobnego obszaru znajdowania się tablicy w celu wyznaczenia pionowych granic.

Dokładny przebieg algorytmu wygląda następująco. Na sam początek obraz zostaje zamieniony w skalę szarości, po czym przeprowadzona zostaje normalizacja obrazu, w celu osiągnięcia jak największego kontrastu pomiędzy jasnymi, a ciemnymi obszarami. Kolejnym krokiem jest rozmycie obrazu przy użyciu filtru gaussowskiego z współczynnikiem $\sigma=2$. Wartość ta została wyznaczona eksperymentalnie podczas przygotowywania i wstępnego testowania algorytmu. Następnym krokiem jest wykrywanie krawędzi. Krawędzie wykrywane są za pomocą połączenia filtrów poziomego i pionowego Sobel'a. Ponieważ zwykłe połączenie tych filtrów powodowało powstawanie wielu zbędnych i niechcianych krawędzi wprowadzono modyfikację. Modyfikacja polega na wykryciu krawędzi pojedynczym filtrem (np. poziomym), następnie przeprowadzenie częściowej binaryzacji, ze sprawdzeniem poziomego i pionowego sąsiada. Jeśli piksel posiada jasność powyżej pewnego progu i jednocześnie posiada sąsiada w pionie lub poziomie, który spełnia ten sam warunek zostaje pozostawiony bez zmian, w przeciwnym razie zostaje zamieniony na piksel koloru czarnego. Próg binaryzacji został ustalony eksperymentalnie podczas tworzenia i wstępnego testowania algorytmu. Po przeprowadzeniu takiego zabiegu osobno dla poziomych i pionowych krawędzi, wyniki filtracji

zostają połączone, a następnie przeprowadzona zostaje kolejna binaryzacja, na tej samej zasadzie jak poprzednio, lecz tym razem w miejsce piksela o jasności większej niż pewien próg zostaje wstawiony piksel koloru białego. Próg podobnie jak poprzednio został wyznaczony eksperymentalnie. Pozwala to na wykrycie najważniejszych krawędzi, jednocześnie minimalizując zbędne krawędzie. Dla usunięcia pojedynczych pikseli użyty został filtr majority. Po takim przygotowaniu obrazu algorytm wykrywa lokalizację poziomych linii, gdzie występuje największa liczba zmian koloru. Następnie dla wykrytego obszaru wyszukiwana jest największe skupisko białych pikseli (w miejscu tablicy powinno ich być najwięcej). Kroki te pozwalają na zlokalizowanie prawdopodobnego miejsca gdzie znajduje się tablica rejestracyjna, niestety pozycja ta zostaje wykryta z różną dokładnością, dlatego wyznaczony obszar zostaje poddany ponownemu poszukiwaniu lokalizacji tablic, tym razem z jak największą dokładnością. Pobrany zostaje tylko wcześniej wyznaczony obszar z oryginalnego wejściowego obrazu, po czym powtórzony jest podobny proces jak poprzednio lecz z innymi parametrami. Wykonane jest kolejno rozmycie filtrem gaussowskim ($\sigma=2.2$ ustalona eksperymentalnie), normalizacja, oraz binaryzacja mieszana. Polega ona na wyznaczeniu progu globalnego, a następnie wyznaczaniu progu lokalnego na najbliższych sąsiadach. Jeśli próg lokalny jest mniejszy niż 1.1 wartości progu globalnego zostaje użyty próg globalny, w przeciwnym razie próg lokalny. Zwiększenie wartości progu globalnego do jego 1.1 wartości została ustalona eksperymentalnie i pozwoliła na duże zwiększenie precyzji lokalizacji tablicy. Następnie ponownie wykrywane są duże ilości zmian w poziomych liniach i skupisko białych pikseli w pionowych liniach. Tym razem bardzo precyzyjnie zostają wykryte liczby i litery na tablicy rejestracyjnej. W celu wyświetlenia całej rejestracji wykryty obszar zostaje poszerzony, w każdym kierunku o pewną, niewielką wartość. W tym momencie algorytm kończy pracę.

Celem programu jest automatyczne wyznaczenie tablicy rejestracyjnej, dlatego integracja użytkownika w parametry algorytmu została ograniczona do minimum, czyli uruchomienia programu. Po uruchomieniu programu wyświetlone zostaje bardzo proste GUI, na którym znajduje się Pole tekstowe, w którym wyświetlany jest czas wykrycia tablicy, oraz przycisk „Save” pozwalający zapisać wykrytą tablicę. Program wczytuje zarówno obrazy w skali szarości, jak i kolorowe. Z powodów ograniczenia algorytmu program działa tylko dla zdjęć o rozdzielczości nie mniejszej niż 600x450 pikseli. Przy próbie uruchomienia programu z obrazem o mniejszej rozdzielczości program wyświetli odpowiednią informację. Dodatkowo, aby przyspieszyć działanie programu, program sprawdza rozdzielczość wczytanego obrazu. Jeśli obraz jest szerszy niż 1024 piksele, program skaluje go do takiej szerokości, zachowując proporcje obrazu. Następnie wykonany jest algorytm opisany powyżej. Końcowym etapem jest wyświetlenie zlokalizowanej tablicy jako nowy obraz i wyświetlenie czasu pracy programu.

3. Testy programu

Podczas tworzenia programu, był on na bieżąco poddawany testom. Przeprowadzono wiele testów, aby wyznaczyć odpowiednie wartości progów i innych wartości, które będą działać dla wszystkich przypadków. Program był tworzony i jednocześnie testowany na ponad 10 obrazach pobranych z internetu o różnych rozdzielczościach, z różnym położeniem tablic na zdjęciach, i różnych tłach. Po osiągnięciu końcowej wersji programu został on przetestowany na kolejnych kilku zdjęciach z internetu i 26 zdjęciach wykonanych przeze mnie telefonem komórkowym (Sony Ericsson Arc S, 8.1 MPix). Program został przetestowany na ponad 40 różnych zdjęciach o różnych rozdzielczościach i różnych jakościach. Osiągnięty wynik testów jest bardzo zadowalający, ponieważ program znalazł 100% poszukiwanych tablic rejestracyjnych, nawet starego typu (biały tekst na czarnym tle). W kilku testowanych przypadkach algorytm wykrył jednak dodatkowo obszary nie będące tablicą

rejestracyjną. Spowodowane jest to małą odpornością algorytmu na refleksy świetlne, jednak należy podkreślić, że są to obszary, które nie są w stanie zakłócić dalszej pracy na wykrytych obszarach, czyli rozpoznawaniu znaków. Poniżej przedstawiono kilka przykładów poprawnie zlokalizowanych tablic, jak również, wspomniane przypadki z nadmiernie wyznaczonym obszarem.



Rysunek 1 Poprawnie zlokalizowane tablice rejestracyjne



Rysunek 2 Zlokalizowane tablice rejestracyjne z dodatkowymi błędnie zlokalizowanymi elementami.

Dzięki użyciu wstępnego skalowania obrazu do szerokości 1024 program osiąga średni czas działania w przedziale od 300-400ms. Poniżej przedstawiono czasy osiągnięte podczas testów:

Time: 421ms	Time: 351ms	Time: 314ms	Time: 294ms	Time: 327ms	Time: 203ms
Time: 424ms	Time: 359ms	Time: 329ms	Time: 289ms	Time: 282ms	Time: 277ms
Time: 351ms	Time: 290ms	Time: 352ms	Time: 340ms	Time: 283ms	Time: 282ms
Time: 339ms	Time: 300ms	Time: 387ms	Time: 300ms	Time: 272ms	Time: 268ms
Time: 286ms	Time: 315ms	Time: 311ms	Time: 305ms	Time: 237ms	Time: 131ms
Time: 296ms	Time: 417ms	Time: 329ms	Time: 306ms	Time: 281ms	Time: 241ms

Tabela 1. Czasy pracy programu zmierzone podczas testów.

Jednym z przeprowadzonych testów było sprawdzenie czasu pracy programu dla różnych rozdzielczości. Ponieważ nie posiadam przy sobie lepszego sprzętu niż telefon, zdjęcia zostały „sztucznie” stworzone poprzez zmniejszanie i powiększanie rozdzielczości oryginalnego zdjęcia. Wyniki jednego z przeprowadzonych testów zostały zamieszczone w tabeli poniżej.

Rozdzielczość obrazu:	Czas pracy programu bez skalowania wstępnego:	Czas pracy programu ze skalowaniem wstępnym:
640x480	Time: 123ms	Time: 161ms
1024x768	Time: 265ms	Time: 231ms
1920x1440	Time: 1018ms	Time: 288ms
3264x2448	Time: 3193ms	Time: 284ms
4896x3672	Time: 8120ms	Time: 257ms
6528x4896	Time: 14988ms	Time: 266ms

Tabela 2. Czasy pracy programu dla różnych rozdzielczości.

Zauważyć należy, że już dla rozdzielczości 1920x1440, czyli niespełna dwukrotnie większej niż używana w programie rozdzielczość czas pracy programu wzrasta prawie pięciokrotnie, co świadczy o poprawności wybranej rozdzielczości pracy programu. Dodatkowo wraz ze wzrostem rozdzielczości algorytm traci na jakości i dokładności (może to być spowodowane sztucznym skalowaniem rozdzielczości).

4. Podsumowanie

Celem niniejszego projektu było stworzenie programu do automatycznej lokalizacji tablic rejestracyjnych na zdjęciach. Program został stworzony wzorując się na algorytmie wykrywania krawędzi i rzutu jasności. Do algorytmu wprowadzono kilka modyfikacji. Modyfikacje te były wynikiem testów, spostrzeżeń oraz własnych opracowań.

W programie nie ma możliwości zmiany parametrów, jest to spowodowane tym, że stworzony program ma działać automatycznie, bez integracji użytkownika. Dla użytkownika wyświetlone zostaje niewielkie i bardzo proste GUI, w którym wyświetlony zostaje czas wykrycia tablicy, oraz znajduje się przycisk pozwalający na zapisanie wykrytej tablicy.

Podczas tworzenia programu przeprowadzono bardzo dużo testów na różnych obrazach z różnymi parametrami. Pozwoliło to na osiągnięcie bardzo zadowalających wyników. Po osiągnięciu końcowej wersji algorytmu, program został przetestowany ponownie na tych samych oraz nowych zdjęciach. Wynik testów był bardzo pozytywny, ponieważ program wykrył wszystkie tablice rejestracyjne. Podczas testów zauważono, że program wykrywa w niektórych przypadkach większy obszar niż sama tablica rejestracyjna, niestety zmiana parametrów algorytmu nie zwiększała precyzji, a zmniejszała ją na innych zdjęciach, gdzie wcześniej tablice były wykrywane poprawnie. Pomimo niedokładności wykrycia tablicy, wykryte obszary jak najbardziej nadają się do dalszej obróbki, ponieważ posiadają zlokalizowaną tablicę rejestracyjną, a pozostałe wykryte elementy nie powinny przeszkadzać w odczytywaniu tablic.

Jednym z ostatnich testów, było uruchomienie programu dla różnych rozdzielczości obrazu. Z uzyskanych wyników przedstawionych powyżej wynika, że pomimo wprowadzenia dodatkowego działania, jakim jest skalowanie, zmniejsza czas pracy programu z 15s do wartości mniejszej niż 300ms.

Podsumowując, śmiało można stwierdzić, iż napisany program został stworzony poprawnie i działa bardzo dobrze, o czym świadczą mogą wyniki wykonanych testów.

Bibliografia

- [1] „Przetwarzanie i rozpoznawanie obrazów w systemach wbudowanych”; Sylwester Madej;
http://marie-www.ee.pw.edu.pl/~czajewsw/studenckie/magisterskie/praca_pM_WE_PW_2012_Madej_Sylwester_196200.pdf

- [2] „Identyfikacja i rozpoznawanie numerów tablic rejestracyjnych pojazdów samochodowych na sekwencjach obrazów wraz z analiza parametrów ruchu”; Marcin Lewkowicz;
<http://www.ee.pw.edu.pl/~czajewsw/studenckie/magisterskie/Lewkowicz.pdf>

- [3] „Filtrowanie obrazów”; Tomasz Lubiński;
<http://www.algorytm.org/przetwarzanie-obrazow/filtrowanie-obrazow.html>