Warszawa, 10.06.2014 r.

Marta Kuzak

**Przetwarzanie Obrazów – Projekt**

**Sprawozdanie**

1. **Założenia projektowe**

* w ramach projektu zrealizowano wykrywanie logo firmy Tesco (*rys. 1.1*).



*Rys. 1.1 Logo firmy Tesco.*

* wczytywane obrazy powinny mieć rozmiar 256 x 256 pikseli
* format obrazów musi być zgodny z formatem dib

1. **Aplikacja**

Program został zaimplementowany z wykorzystaniem aplikacji DIBLOOK, z której korzystano na zajęciach laboratoryjnych z Przetwarzania Obrazów.

* Język: C++
* Środowisko programistyczne: Microsoft Visual Studio 2012 (32-bit)

W celu wykrycia logo należy uruchomić program i korzystając z interfejsu graficznego wybrać testowany obraz (*File -> Open*). Następnie należy wybrać w menu *Operation -> Block*. Algorytm zacznie działanie.

1. **Opis algorytmu**
2. Filtracja filtrem medianowym
3. Konwersja składowych RGB opisujących obraz na składowe HSV
4. Utworzenie obrazu binarnego

Wszystkie piksele, których wartości składowych H i S mieszczą się w ustalonym przedziale przypisywany jest kolor biały, pozostałym – czarny.

Logo Tesco składa się z obiektów w dwóch kolorach. W trakcie sprawdzania wartości pikseli tych elementów na różnych obrazach, zaobserwowano, że elementy niebieskie bardzo różnią się pomiędzy sobą. Ustawienie odpowiednio szerokiego marginesu, ale program poprawnie je wyodrębniał, skutkowało obecnością w obrazie binarnym wielu obiektów, które de facto były składnikami tła. Z tego powodu zdecydowano, że na początku aplikacja będzie wyszukiwać napisu „TESCO”.

1. Erozja
2. Dylacja
3. Nadanie etykiet wyodrębnionym obiektom
4. Obliczenie współczynników i identyfikacja obiektów

Identyfikacja obiektów następuje z wykorzystaniem następujących współczynników: M1, W3, W7.

Współczynnik M1 pozwala na odrzucenie większej liczby zbędnych obiektów, które pozostały po segmentacji. Ponieważ litery w logo Tesco nie różnią się szczególnie rozmiarem (nie różnią się bardzo prostokątami, które można na nich opisać), wartości tego współczynnika dla nich są do siebie dość podobne i nie wystarczają do odróżnienia jednej litery od drugiej.

Wprowadzenie współczynnika W3 pozwoliło na prawie całkowitą możliwość odróżnienia liter od siebie. Konflikt znajdował się jedynie w przypadku liter S i O. Został on rozwiązany poprzez użycie współczynnika W7, który wskazuje na stosunek najmniejszej do największej odległości konturu obiektu od jego środka ciężkości.

Poniższa tabela (*tab. 2.1*) przedstawia zakresy wartości wyżej opisanych współczynników dla kolejnych liter.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **T** | | **E** | | **S** | | **C** | | **O** | |
| **min** | **max** | **min** | **max** | **min** | **max** | **min** | **max** | **min** | **max** |
| **M1** | 0.28 | 0.36 | 0.25 | 0.33 | 0.24 | 0.32 | 0.34 | 0.42 | 0.29 | 0.37 |
| **W3** | 0.75 | 0.93 | 1.04 | 1.23 | 1.35 | 1.70 | 1.01 | 1.41 | 1.32 | 1.61 |
| **W7** | 0.00 | 0.01 | 0.00 | 0.01 | 0.00 | 0.01 | 0.12 | 0.28 | 0.34 | 0.52 |

W przypadku rozpoznawania napisu konieczne było również sprawdzenie:

* Czy pojawiają się one w odpowiedniej kolejności
* Czy znajdują się one w odpowiedniej odległości od siebie (liczono odległości pomiędzy środkami ciężkości w zależności od rozmiarów liter)
* Czy wymiary liter potencjalnie tworzących napis są odpowiednio – różnią się jedynie w dozwolonym zakresie

1. Narysowanie prostokąta wokół wykrytego logo

Przykładowy obraz wygenerowany przez aplikację przedstawia *rys.* *2.1*



*Rys. 2.1 Obraz z defekowanym logo*

1. Testowanie