

# Przetwarzanie Obrazów

## Sprawozdanie z projektu

### 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

### 2. Aplikacja

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

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

Najwięcej zmian dokonywano w pliku *dibdoc.cpp*. Funkcja rozpoznawania logo to `void CDibDoc::ConvertToGrayImage( CView *view )`.

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

### 3. Opis algorytmu

#### 1) Filtracja filtrem medianowym

Zastosowanie filtru medianowego ma na celu zmniejszenie poziomu szumu w obrazie.

2) Konwersja składowych RGB opisujących obraz na składowe HSV

Dzięki konwersji obrazu na składowe HSV możliwa będzie segmentacja na podstawie koloru niezależnie od jasności pikseli.

3) 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 (czerwone litery i pod spodem niebieskie figury). W trakcie sprawdzania wartości pikseli tych elementów na różnych obrazach, zaobserwowano, że elementy niebieskie bardzo różnią się pomiędzy sobą. Ustawienie odpowiednio szerokiego marginesu, aby program poprawnie je wyodrębniał, skutkowało w niektórych obrazach obecnością w obrazie binarnym wielu obiektów, które de facto były składnikami tła.

Z tego powodu zdecydowano, że algorytm znajdowania logo będzie podzielony na dwa następujące po sobie etapy: najpierw wykrycie napisu TESCO, następnie niebieskich elementów pod nim.

4) Erozja

Erozja pozwala na rozłączenie obiektów, które po wstępnej segmentacji łączą się ze sobą małym obszarem oraz usunięcie małych obiektów, które nie powinny znaleźć się w obrazie binarnym.

5) Dylacja

W wyniku erozji obiekty mogą być zniekształcone, mogą pojawiać się w nich dziury, robią się one „wychudzone”. Dylacja naprawia te defekty.

6) Nadanie etykiet wyodrębnionym obiektom

Obiekty muszą zostać ponumerowane – każdy z nich ma odrębną, unikatową etykietę.

7) Obliczenie współczynników i identyfikacja obiektów

- Napis TESCO

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

Poniższa tabela (*tab. 3.1*) przedstawia zakresy wartości wyżej opisanych współczynników dla kolejnych liter.

	T		E		S		C		O	
	min	max	min	max	min	max	min	max	min	max
<b>M1</b>	0.28	0.36	0.25	0.33	0.24	0.32	0.34	0.42	0.29	0.37
<b>W3</b>	0.75	0.96	1.04	1.29	1.35	1.70	1.01	1.41	1.32	1.61
<b>W7</b>	0.00	0.01	0.00	0.01	0.00	0.01	0.12	0.28	0.34	0.54

*Tab. 3.1 Wartości współczynników M1, W3 i W7 dla liter napisu TESCO.*

W przypadku rozpoznawania napisu konieczne było również sprawdzenie:

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

- Niebieskie elementy

Do identyfikacji niebieskich elementów logo brane są pod uwagę tylko obiekty, które znajdują się w odpowiedniej odległości od napisu TESCO (i znajdują się poniżej tego napisu). Liczone są dla nich współczynniki M1 i W3 (tab. 3.2).

	min	max
<b>M1</b>	0.23	0.36
<b>W3</b>	0.21	0.49

*Tab. 3.3 Wartości współczynników M1 i W3 dla niebieskich elementów logo.*

Ponadto sprawdzane jest:

- Czy te obiekty są do siebie zbliżone wielkością (stosunek ich pól powinien być zbliżony do 1)
- Czy znajdują się względem siebie w odpowiedniej odległości

#### 8) Narysowanie prostokąta wokół wykrytego logo

Pozytywne zakończenie algorytmu skutkuje narysowaniem zielonego prostokąta wokół logo.

#### 4. Testowanie

Aplikację testowano za pomocą obrazów testowych znajdujących się w katalogu *test*. Wczytywano kolejne obrazy i sprawdzano rezultaty działania programu. Jeżeli logo zostaje wykryte, wokół niego zostaje narysowany zielony prostokąt. Przykładowy obraz przed i po rozpoznaniu logo przedstawia *obraz 4.1*.



*Obraz 4.1 Po lewej stronie – obraz wejściowy algorytmu, po prawej – wyjściowy.*