Finding Pepsi

Paweł Rybak

Program ma na celu wykrywać loga pepsi na obrazie wejściowym. Ścieżka do obrazu jest podawana jako argument wywołania programu. Po przetworzeniu, wyświetlana jest wersja obrazu z wykrytymi elementami obrysowanymi zielonym prostokątem.

Kompilacja

Do skompilowania programu wymagana jest biblioteka OpenCV. Kompilację należy przeprowadzać z użyciem standardu C++17.

Działanie programu

Program wykrywa logo poprzez znalezienie osobno czerwonej oraz niebieskiej częsci loga, a następnie dobrania dwóch części znajdujących się odpowiednio blisko. Wyodrębnienie poszczególnych części składa się z utworzenia maski obrazu na podstawie wyodrębnienia koloru oraz progowania, a następnie podzielenie maski na obszary ciągłe i wybranie tych obszarów, które posiadają odpowiednie parametry. Finalnie znalezione elementy są parowane na podstawie odległości. Dokładne kroki, oraz dobrane parametry można zobaczyć w pliku *main.cpp*.

Użyte algorytmy

Wszelkie algorytmy użyte w tym projekcie zostały zaimplementowane od podstaw. Ich implementacje znajdują się wśród plików projektu.

Tworzenie maski

Funkcja *filterColor*

Funkcja ustawia wartość każdego piksela według wzoru:

```
px = col(px) - lum(px)
```

gdzie `lum(px)` oznacza jasność danego piksela, a `col(px)` wartość danego piksela w kanale koloru, którego poszukujemy.

Progowanie

Progowanie obrazu pozwoliło wyodrębnić elementy o interesującym poziomie jasności, z obrazu wynikowego poprzedniej funkcji.

Otwarcie i domknięcie

Funkcja otwarcia pozwoliła na oddzielenie interesującego fragmentu od tła. Szczególnie istotne było to w przypadku niebieskiego fragmentu loga, które potrafiło zlewać się z niebieskim tłem etykiety. Operacja domknęcia wyeliminowała z elementów zakłócenia, takie jak światło odbijające się od etykiety.

Wodrębnienie obszarów

Funkcja *getSegments*

Funkcja, po znalezieniu interesującego piksela (np. białego), zamalowuje go na szaro i aplikuje ten sam algorytm dla każdego sąsiada. Po zamalowaniu wszystkich pikseli obszaru, jest on wyodrębniany jako osobna maska, a region na oryginalnym obrazie zamalowany na kolor tła. Funkcja ta zwraca wektor masek, gdzie każda reprezentuje osobny region.

Współczynnik Malinowskiej, M3 i M7

Dla każdego obszaru był liczony współczynnik malinowskiej, a w przypadku segmentów czerwonych, również współczynniki M3 i M7. Na podstawie wartości tych współczynników, część masek była odrzucana. Współczynniki te zostały dobrane osobno dla każdej klasy elementów (czerwonej częsci i niebieskiej częsci loga) metodą eksperymentalną.

Środek ciężkości

Aby sprawdzać odległość między elementami, zaimplementowana została funkcja znajdująca środek ciężkości kształtu. Dzięki temu liczenie odległości między dwoma kształtami można sprowadzić do liczenia odległości między dwoma punktami.

Odległość względna

Dla każdej pary z listy elementów niebieskich i listy elementów czerwonych liczona była odległość względna według wzoru:

```
rDist = dist(sh1, sh2) / sqrt(field(sh1) + field(sh2))
```

gdzie `sh1` oraz `sh2` oznaczają porównywane kształty, funkcja `dist` liczy odległość między środkami ciężkości elemetów, a `field` liczy powierzchnie elementu. Odległość jest dzielona przez sumę pól pod pierwiastkiem, ponieważ przy przybliżaniu obrazu pole rośnie proporcjonanie do kwadratu powiększenia, a odległość liniowo. Zakres odległości między elementami loga został zbadany eksperymentalnie.

Wnioski

Największym zaskoczeniem był dla mnie czas działania programu. Faktem jest, iż podczas tworzenia programu wydajność nie była kluczowa, jednak czas przetwarzania obrazu o wielkości ok. 1Mpx był zaskakująco długi.

Kolejnym niespodziewanym aspektem był duży wpływ błędów przy akwizycji na efekt działania. Błędy takie jak zbyt mała rozdzielczość zdają się być oczywiste, jednak także niepoprawne oświetlenie, które dla oka zdawało się być nieznaczące potrafiło mocno utrudnić pracę programu. Pomagały tutaj operacje otwarcia i domknięcia, jednak nie zawsze były w stanie poprawić błędy akwizycji.

Przy filtracji wyodrębnionych segmentów obrazu bardzo dobrze sprawdził się współczynnik malinowskiej. Przy mniejszych elementach dawał coraz mniej przewidywalne rezultaty, dlatego poza nim użyte zostały współczynniki M3 i M7. Mimo to największą pomocą był fakt, iż logo to składa się z kilku różnokolorowych segmentów. Pozwoliło odrzucić segmenty, które przeszły poprzednie filtracje, a które znajdują się daleko od jakichkolwiek segmentów o odpowiadającym kolorze.