

# POBR Projekt

Michał Przyłuski — 198361

10 czerwca 2008

## 1 Cel projektu

Celem projektu było rozpoznanie logo firmy *Sun Microsystems*.



Z analitycznego punktu widzenia, logo to składa się z 8 podobnych obiektów, ułożonych na planie rombu, oraz liter *Sun* po jego prawej stronie.

## 2 Realizacja

Zadanie zostało zrealizowane, w oparciu o zaprezentowany na zajęciach laboratoryjnych szkielet — aplikację *diblook*. W jeden z jej funkcji, został zaimplementowany algorytm mający na celu rozpoznanie i zlokalizowanie powyższego logo na dostarczonym przez użytkownika pliku w formacie *dib*. W obecnej wersji maksymalny rozmiar obrazka wynosi 900 x 900 pikseli, i wynika ze sztywno alokowanych tablic — buforów na kolejne fazy przetwarzania.

## 3 Zastosowany algorytm

Rozpoznanie znaczka przebiega w kilku etapach.

1. **Negacja** wszystkich kolorów w obrazku. Zaszłość z wersji rozwojowej programu, łatwiej wyobrazić sobie rozpoznawanie białych obiektów na czarnym tle niż odwrotnie.

2. **Filtr medianowy** — o rozmiarze  $3 \times 3$ , wybierany element o numerze 4 (numeracja przebiega od 0 do 8), zatem jest to faktycznie mediana. Działanie takie ma na celu redukcję szumów na obrazku, i lekkie jego „wygładzenie”. Filtr medianowy, bowiem, wyeliminuje gwałtowne zmiany jasności w danym punkcie, jeśli nie są one poparte odpowiednimi zmianami punktów sąsiednich.
3. **Progowanie**, wszystkie punkty o kolorach odpowiadających rozpoznawanemu logo, są wyznaczone na białą.
4. **Rozrost**. W tym przebiegu przez obrazek, podobnie jak na etapie filtracji medianowej, są przetwarzane bloki  $3 \times 3$  piksele. Tym razem jednak wybierany jest piksel o największej wartości funkcji jasności.
5. Kolejny etap **progowania**, teraz już zero-jedynkowy. Logo, po rozroście, białe, kolorowane na czarno, cała reszta na białą. Efekt końcowy tej fazy to właśnie czarne fragmenty na białym tle.
6. Kluczowym etapem w rozpoznaniu obrazu jest **segmentacja**. Celem jest uzyskanie listy pojedynczych, wyodrębnionych *obiektów* na podstawie jednoznacznie pokolorowanych pikseli z poprzedniego etapu. W tym celu przetwarzane są kolejne punkty (czarne) obrazu. Z każdego punktu sprawdzani są jego wszyscy sąsiedzi. Informacja o następnych punktach przechowywana jest w liście struktur Punkt. Ponadto, odwiedzone punkty oznaczane są odmiennym kolorem.

Na tym kończy się „ogólne” przetwarzanie, któremu były poddawane wszystkie punkty wejściowego obrazka, po kolei. Poniższe operacje będą odnosić się do wysegmentowanych powyżej obiektów, do każdego z nich niezależnie. Iterujemy się bowiem po liście obiektów.

1. Na pustym obrazku (białe tło) nakładany jest rozważany obiekt (czarny).
2. W toku przeglądania jego pikseli wyznaczane są m. in. współczynniki  $M_1$  oraz  $M_7$ . A także inne pewne parametry opisane w **Podstawy teoretyczne**.
3. Na podstawie wyznaczonych współczynników obiekt jest klasyfikowany jako detal,  $S$ , lub coś innego.

Ostatecznie, wszystkie obiekty w obu interesujących nas klasach, są kolorowane, wyznaczany jest ich uśredniony środek, i na tym rozpoznawanie się kończy.

System udziela też odpowiedzi tekstowej, potwierdza znalezienie logo w przypadku gdy stwierdzono między 7 a 10 detali oraz od 1 do 2  $S$ .

## 4 Podstawy teoretyczne

Do rozpoznania obiektów zostały użyte współczynniki kształtu  $M_1$  oraz  $M_7$ . Wydaj się, że zachowują się one dość dobrze w odpowiedzi na umiarkowane zmiany położenia i rozmiaru obiektów.

Współczynnik  $M_1 = \frac{M_{20} + M_{02}}{m_{00}^2}$ , gdzie  $m_{00}$  oznacza pole powierzchni, zachowuje się bardzo stabilnie dla rozważanych obiektów. Nie wykazuje on istotnych statystycznie wahań. (Dla wartości rzędu  $0.220 \pm 0.020$  w przypadku najbardziej zróżnicowanego, ze względu na swoją liczbę wystąpień, detalu w rombie).

Współczynnik  $M_7$  ma pewne negatywne cechy. Jest on dość wrażliwy na nawet subtelne zmiany krawędzi obiektu. Zmiany takie mogą powstać na etapie erozji, w wyniku przetwarzania obrazka, na którym logo znajdowało się na pewnej powierzchni o istotnej fakturze. W drodze eksperymentu ustalono akceptowalny  $M_7 = 8.2 \times 10^{-3} \pm 0.75 \times 10^3$ .

Dla detalu w rombie uzyskano dobrą specyficzność rozpoznania. Niestety jako jego fragment rozpoznawana jest także litera *y* z podpisu *Microsystems* pod logo. Istotnie, litera ta ma bardzo zbliżone parametry  $M_1$  oraz  $M_7$  do pożądaných obiektów.

Szczęśliwie, w obliczu znacznej liczby poprawnych detali, silnie skondensowanych w obszarze rombu, pojedynczy obiekt zewnętrzny nie wpływa negatywnie na jakość rozpoznania. Jest to możliwe, gdyż wszystkie obiekty klasy „detal” mają wyznaczany środek (geometryczny), który następnie jest uśredniany. W efekcie dochodzi jedynie do nieznacznego przesunięcia tego środka w dół i w prawo od środka rombu. Nie ma to jednak istotnego wpływu na ostateczną jakość rozpoznania.

Litera *S* wyznaczona jest z większą specyficznością. Akceptowane dla niej wartości parametrów to  $M_1 = 0.5025 \pm 0.0115$  oraz  $M_7 = 3.925 \times 10^{-2} \pm 0.2 \times 10^{-2}$ .

W toku realizacji projektu rozważone zostały także inne możliwe kombinacje współczynników. Szczególnie kuszące wydawało się zastosowanie ilorazu pola i obwodu. Niestety okazało się to nie wystarczające do jednoznacznego rozpoznania obiektów różnych rozmiarów.

Ponadto, w tym samym czasie gdy zbierane są dane do wyżej wymienionych współczynników kształtu, wyznaczane są wierzchołki najmniejszego prostokąta opisującego badany zbiór. Na podstawie wyznaczonych danych obliczane jest położenie *geometrycznego* środka figury. Znaleziony środek jest dodawany do zbioru środków, w zależności od tego do jakiej figury należy, detali lub potencjalnych „eSów”.

W dalszym kroku wszystkie środki z danej klasy są uśredniane. Podejście takie pozwala wyznaczyć przybliżony środek rombu ograniczającego osiem detali. Dzięki temu, nawet przypadkowe — błędne — wykrycie innego fragmentu logo jako detalu (sytuacja taka ma czasem miejsce z literą *y*) ma ograniczony wpływ na rozpoznanie położenia rombu.

Podjęte zostały próby wyeliminowania jej wpływu poprzez odrzucenie obiektów o zbyt małej powierzchni lub obwodzie, jednakże swoją materializację znalazły one dopiero w wersji 2.

## 5 Problemy realizacyjne

Podstawowym problemem okazało się wstępne odizolowanie logo od tła. Wynika to z faktu, iż logo to występuje w wielu kolorach, i trudno jest podać dobre kryterium na jego rozpoznanie. W szczególności, 3 testowe obrazki, które pod względem kolorystycznym wydawały mi się dość podobne, używały następujących kolorów jako główny kolor logo. (53, 98, 119), (77, 102, 169) i ok. (65, 70, 100).

Jako wartości graniczne przyjęto zatem  $x < (115, 120, 195)$ .

Jak widać graniczna wartość jest dość wysoka, nie jest to ściśle wydzielenie obiektów. Jednakże możemy tak postąpić, gdyż wstępna selekcja podejrzanych obiektów na podstawie koloru zostanie następnie zweryfikowana przez analizę ich kształtów.

Istotnym ograniczeniem jest faktura powierzchni, z której jest zbierane zdjęcie z logo. Jeśli powierzchnia nie jest dostatecznie gładka, dojdzie do dość znacznego rozrostu krawędzi detali wchodzących w skład rombu. Gdy tak się stanie, w wyniku rozrostu, a następnie zastosowanej metody segmentacji, może nastąpić dość tragiczne w skutkach zlanie (części) detali w jedną bezkształtną figurę.

Dodatkową trudnością w rozpoznaniu małego logo, w przypadku plików w formacie JPEG jest fakt, że dochodzi tam do istotnych odchyień kolorystycznych. Wynikają one z tego, iż JPEG zapisuje informacje w blokach 8x8, czyli w pewnym stopniu tło (z reguły jasne, ale zawsze inne niż logo) wpływa na odchylenie koloru logo w kierunku tła. Powoduje to istotne wahania funkcji jasności, oraz poszczególnych składowych w okolicy krawędzi.

W przypadku dużego logo, problem ten ma umiarkowane znaczenie, jednakże dla małego logo, okazuje się, że kolorystycznie jest ono bardzo niejednolite.

## 6 Wersja 2

Powyżej została zaprezentowana pierwsza wersja systemu. Jednakże, miała ona istotne ograniczenie, wymagała dość dużego logo. W praktyce obrazek o wymiarach rzędu 600x600 był pozytywnie rozpoznany, ale zdjęcie o takich wymiarach już nie. Konieczne było podanie do programu zdjęcia o rozmiarze 900x900.

Przyczyną takich problemów było, najprawdopodobniej, przedstawione w poprzednim punkcie, zjawisko zlewania się poszczególnych detali.

Aby temu zaradzić, zmienony został algorytm przetwarzania. Zmieniony został parametr rozrostu. Zamiast wybierania (jak w oryginale) ostatniego, dziewiątego piksela po posortowaniu, wybierany jest czwarty. Efekt takiego zabiegu jest taki, że zamiast rozrastać się, obiekty kurczą się! Dzięki temu wybiegowi, nawet mniejsze logo staje się wykonalne, gdyż nie dochodzi do niekorzystnego zjawiska zlewania się fragmentów.

Pewnego dostosowania wymagały współczynniki. Obecnie,  $M_1 = 0.2575 \pm 0.0125$ ,  $M_7 = 10.6 \times 10^{-3} \pm 0.95 \times 10^{-3}$  dla detalu, oraz dla dużego S:  $M_1 = 0.56 \pm 0.02$  oraz  $M_7 = 4.95 \times 10^{-2} \pm 0.375 \times 10^{-2}$ . Uzyskano zatem zadowalającą specyficzność tych współczynników.

Niestety, w obliczu nowych współczynników, kolejna mała litera została zaklasyfikowana jako poszukiwany fragment. Tym razem małe s do klasy S.

W przypadku poprzednio wykrywanego y problem ten został zignorowany, gdyż duża liczność poprawnych detali, gwarantowała akceptowalne położenie środka rombu.

Konieczność taka zaszła teraz, gdyż wykrycie jednego ekstra obiektu w klasie S znacznie przesuwał środek tej klasy.

Litera S jest rozpoznawana tylko gdy jej pole przekracza 200 pikseli. Jest to rozsądny próg, gdyż właściwe S jest naprawdę duże. Zabieg ten ma na celu wyeliminowanie sytuacji gdy małe s zostałyby

rozpoznane jako fragment logo. Kolejnym możliwym rozwinięciem tej idei, mogło by być wyznaczenie średniej ważonej, o wagach zależnych od pola obiektu.

Największym sukcesem wersji drugiej, jest poprawne zlokalizowanie logo w przypadku użycia oryginalnego (tylko zmniejszonego do wysokości 900 pikseli) zdjęcia z aparatu.

## 7 Efekty

System poprawnie rozpoznaje odpowiednie elementy logo dla *rysowanego* logo w wielu konfiguracjach. Tak długo jak logo będzie dość duże, czyli segmentacja się powiedzie, współczynniki kształtu poprawnie wyznaczą kluczowe elementy.

Udało się także zredukować o kwant wymagania rozmiarowe dla autentycznego *zdjęcia*. Pierwsza wersja nie była w stanie rozpoznać logo ze zdjęcia o szerokości 600 pikseli, działała poprawnie dla 900. Wersja druga, z „negatywnym rozrostem”, daje poprawną odpowiedź dla mniejszego obrazka.

## 8 Możliwe rozszerzenia

Wartościowym dodatkiem byłoby realizacja wyrównania histogramu — zapewnienie dodatkowego kroku przetwarzania wstępnego odpowiadającego „auto-level” ze znanych programów graficznych. Działanie takie było konieczne w przypadku jednego ze zdjęć, którego balans bieli daleko odbiegał od oczekiwanego.

W przypadku oryginalnego zdjęcia rozpoznawany jest dodatkowo duży obszar po lewej stronie. Wynika to z faktu, iż tło (ściana, teoretycznie biała) została uwieczniona na fotografii przez niskiej klasy aparat, jako niebieska. Rozsądne zatem byłoby wprowadzenie górnego ograniczenia na pole rozpoznawanego detalu.