POBR - Rozpoznawanie kostki Rubika

Mateusz Boryń

14 czerwca 2010

1 Założenia

1. Program ma za zadanie zlokalizować jedną ścianę kostki Rubika i określić kolory kafelków na tej ścianie.
2. Na kostce Rubika znajdują się kafelki o kolorach: czerwony, pomarańczowy, żółty, zielony, niebieski, biały.
3. Kafelki są oddzielone ciemną linią (szczelina w kostce Rubika)
4. Na ścianie kostki Rubika musi być widoczne co najmniej 7 kafelków

2 Filtrowanie

W celu zredukowania zakłóceń wprowadzonych przez tor akwizycji (szum typu śalt & pepper") zastosowałem filtr medianowy. Maska filtru ma rozmiar 3×3.

Operacja filtrowania jest realizowana przez funkcję medianBlur() z biblioteki OpenCV.

3 Klasyfikacja kolorów

Program przyjmuje obraz wejściowy w przestrzeni barw RGB. Ze względu na fakt, że w tej przestrzeni barwa i jasność są zawarte we wszystkich trzech składowych, identyfikacja koloru jest bardzo trudna. Dlatego stosuję zamianę przestrzeni barw z RGB na HSV, gdzie barwa jest w jednej składowej, nasycenie i jasność w dwóch pozostałych. Przeprowadziłem również próby z wykorzystaniem przestrzeni barw YCrCb, W przestrzeni tej dwie składowe odpowiadają za barwę i jedna za jasność. Wynika stąd, że w przestrzeni YCrCb trudniej jest analizować kolory niż w przestrzeni HSV.

W przypadku idealnej reprezentacji kolorów w przestrzeni HSV poszczególne kolory (z wyjątkiem białego) mają następującą wartość składowej Hue (od 0 do 359):

1. czerwony: 0
2. pomarańczowy: 20
3. żółty: 60
4. zielony: 120
5. niebieski: 240

Składowa Saturation dla wszystkich wyżej wymienionych kolorów została dobrana na od 100 do do 255 (w skali 0..255). Dla koloru białego składowa Saturation zawiera się w przedziale [0; 60], natomiast składowa Hue nie ma znaczenia.

Kolory czerwony i pomarańczowy są do siebie bardzo zbliżone. W celu odróżnienia tych dwóch kolorów należy przyjąć arbitralnie oddzielający je próg. Dla obrazów uzyskanych z różnych kamer próg ten może być inny.

Przed klasyfikacją przygotowywane są trzy tabele LUT dla każdej składowej H, S, V zawierające po 256 elementów (dla składowej H przedział dopuszczalnych wartości to [0;180), żeby można było używać liczb całkowitych 8-bitowych, dla S i V to [0;255]). Każdy bit elementu LUT odpowiada jednej klasie koloru. Tabele LUT są indeksowane poprzez HSV danego piksela. Klasa koloru jest określana jako iloczyn bitowy trzech wyznaczonych w ten sposób elementów. Należy zadbać o to, aby klasy kolorów nie nakładały się. Jeśli piksel nie zostanie zaklasyfikowany do żadnego znanego koloru, to jego klasa to 0. Jeśli piksel zostanie zaklasyfikowany jako jakiś kolor to jego klasa to 1«(indeks koloru).

4 Segmentacja

Algorytm segmentacji zaimplementowałem w następujący sposób (algorytm pracuje na obrazie jednokanałowym, zawierającym klasy kolorów jako wartości poszczególnych pikseli):

1. Przeglądaj obraz oryginalny wiersz po wierszu, aż do znalezienia piksela P_0, który ma znany kolor i nie jest jeszcze przydzielony do segmentu.
2. Odłóż współrzędne punktu P_0 na stos.

3. Zaznacz punkt P_0 jako przydzielony do segmentu.

4. Pobierz ze stosu punkt P.

5. Dodaj punkt do obrazu segmentu.

6. Dla każdego z sąsiadów punktu P: jeżeli jest tego samego koloru, co P_0, to odłóż go na stos i zaznacz jako przydzielony do segmentu.

7. Jeżeli stos jest pusty - zakończ, wpp. idź do punktu 4.

5 Rozpoznawanie kształtów

Po dokonaniu segmentacji mamy zazwyczaj dużo segmentów o różnych kształtach. Należy odfiltrować listę segmentów pod względem ich pola powierzchni i kształtu.

Rozpoznawanie kształtów odbywa się na podstawie wartości niezmienników momentowych M_1 i M_7. Dla kwadratu, wartości tych współczynników zostały dobrane jako $M_1 \in (0.162; 0.175)$ oraz $M_7 \in (6.5 \cdot 10^{-3}; 7.5 \cdot 10^{-3})$.

6 Rozpoznawanie obiektu

Mając daną listę segmentów będących kwadratami, trzeba znaleźć strukturę przypominającą ściankę kostki Rubika. W tym celu dla każdego elementu z listy odfiltrowanych segmentów (oznaczam go jako element centralny) przeglądamy listę segmentów. Szukamy takich segmentów, które mają wielkość zbliżoną do segmentu centralnego oraz znajdują się w odpowiedniej odległości od segmentu centralnego. Następnie segmenty znalezione w pobliżu segmentu centralnego są sortowane wg kąta, jaki tworzą z segmentem centralnym. W ten sposób można znaleźć orientację kostki Rubika.

7 Wnioski

W analizowanym obrazie kostki Rubika kafelki muszą być wyraźnie oddzielone, inaczej zamiast dwa segmenty zleją się w jeden, który będzie miał niezmienniki znacznie odbiegające od zadanych.

Dla znacznie zdeformowanego obrazu kostki Rubika (np. poprzez zniekształcenia geometryczne wprowadzane przez optykę kamery) poszczególne znacznie odbiegają kształtem od kwadratu i nie mogą być poprawnie zaklasyfikowane.

Podczas wykonywania zdjęć pojawiały się odblaski, które uniemożliwiały poprawne rozpoznanie koloru.