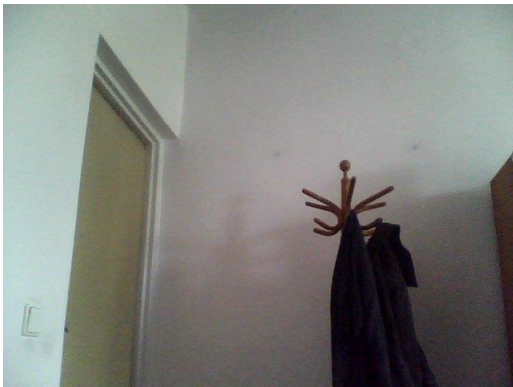


Detekcja ruchu

Hanna Palonek

Zadanie do wykonania:

W ramach zajęć laboratoryjnych należało dokonać detekcji ruchu w najprostszym przypadku tzn. gdy ma się do dyspozycji jedynie dwie klatki obrazu. W opisywanym przykładzie na pierwszej klatce znajdował się obraz pokoju, a na drugiej dłoń na tle tego samego pokoju. Ostatecznym wynikiem pracy miał być obrys ręki, która pojawiała się na drugiej klatce.



Rys 1. Tło bazowe



Rys 2. Analizowana klatka



Rys 3. Obraz ostateczny

Etapy wykonania:

1. Odjęcie dwóch obrazów

Pierwszą wykonaną czynnością było odjęcie obrazu z dłonią od obrazu tła, przy pomocy funkcji `imsubtract(obraz1, obraz2)`, w celu uzyskania obrazu jedynie z dłonią. Jednak wynik tej funkcji nie był satysfakcjonujący, uzyskano jedynie fragment dłoni w kształcie wieszaka. Po zamianie odjemnej i odjemnika uzyskano obraz dłoni z dużym ubytkiem w kształcie wieszaka. Zdecydowano, że najlepszym wyjściem będzie dodanie obu uzyskanych różnic poprzez funkcję `imadd(różnica1, różnica2)`.



Rys 4. Wynik pierwszego odejmowania



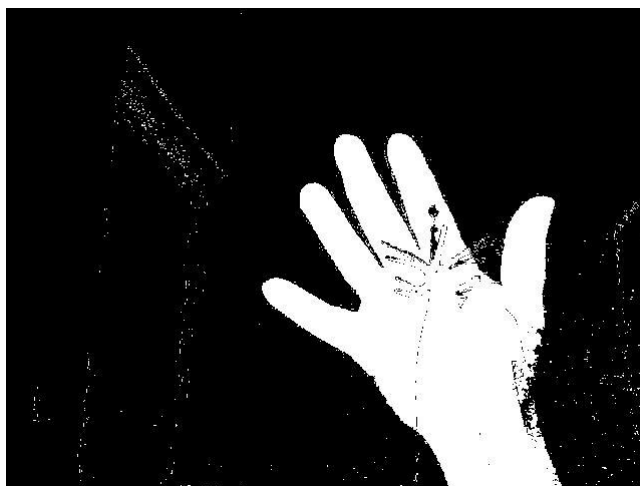
Rys 5. Wynik drugiego odejmowania



Rys 6. Obraz po dodaniu obu różnic (który będzie poddany kolejnym przeróbką) [etap1-reka.jpg]

2. Binarizacja obrazu

Kolejnym etapem obróbki obrazu była jego binaryzacja. Wykorzystano prostą funkcję matlaba `im2bw(obraz, próg)`, a próg dobrano metodą „prób i błędów” na 0,05. Przy progu wyższym pozostawało dużo mniej szumów, ale tracono część dłoni. Przy niższym progu zaś dłoń była widoczna niemal w całości, ale pozostawione elementy w tle ciężko było usunąć bez uszczerbku na obiekcie.



Rys 7. Obraz po binaryzacji z progiem 0.05 [etap2-binarny.jpg]

3. Oczyszczenie obrazu

W celu uzyskania jedynie obrazu dłoni dokonano kilku operacji usuwających szum i zarysy innych przedmiotów. Najpierw użyto filtra medianowego z macierzą [3, 3] - `medfilt2(obraz, [3 3])`, następnie operacji morfologicznej - erozji z ustalonym kształtem maski na kwadrat, a wielkością na 2 jednostki – `imerode(obraz, se) -> se = strel('square', 2)`. Po usunięciu części obrazu przez erozję ponownie użyto filtru medianowego.



Rys 8. Obraz po pierwszym użyciu filtra medianowego [etap3-szum.jpg]



Rys 9. Obraz po wszystkich operacjach 'oczyszczenia' [etap4-morfologia.jpg]

4. Wypełnienie dziur

W uzyskanym obiekcie pozostało wiele luk, aby uzyskać jak najdokładniejszy obrys dłoni należało wcześniej wypełnić cały obszar. Program matlab dysponuje specjalną funkcją do wypełniania dziur mianowicie – `imfill(obraz, 'holes')`. Po jej wykonaniu uzyskano niemal pełny obraz dłoni, ubytki wynikają z progowania przy binaryzacji.



Rys 10. Wypełniony obraz dłoni [etap5-wypelnianie.jpg]

5. Wykrywanie krawędzi

Niemal ostatnim etapem przetwarzania obrazu było wykonanie obrysu uzyskanego obiektu. Wykorzystano operację wykrywania krawędzi dwoma metodami wg Sobel'a, oraz wg Canny'ego . Lepszy rezultat osiągnięto dzięki algorytmowi Canny'ego – `edge(obraz, 'canny')`



Rys 11. Obrys dłoni uzyskany metodą Canny'ego [etap6-krawedziecannego.jpg]

6. Nałożenie krawędzi na obraz pierwotny

Obrazy należało doprowadzić do postaci gdzie na obraz dłoni w pokoju w skali szarości nałożony został obrys dłoni. Operacja dodania do siebie dwóch obrazów może jednak być wykonana jedynie gdy oba obrazy mają ten sam tym i rozmiar. W tym celu zamieniono binarny obraz z krawędziami na skalę szarości – `uint8(obraz)*255` , do tej samej postaci doprowadzono zdjęcie dłoni w pokoju, lecz inną metodą - `rgb2gray(obraz)`



Rys 12. Ostateczny wynik, obrys dłoni na pierwotnym zdjęciu [ostateczny.jpg]

Wnioski:

Do wykonania ćwiczenia korzystano z programu matlab, wszystkie potrzebne metody i funkcje zostały odnalezione na stronie internetowej programu. Cały stworzony kod oraz obrazy poszczególnych etapów pracy zostały dołączone do sprawozdania.

Najtrudniejszym etapem przetwarzania obrazu był dobór odpowiednich parametrów przy binaryzacji i oczyszczaniu dłoni. Sprawdzono wiele opcji metodą 'prób i błędów', na przykład inne operacje morfologiczne, inne kształty przy ich użyciu. Głównym celem było zlikwidowanie niepotrzebnych elementów przy jednoczesnym zachowaniu jak największej ilości samej dłoni.

Uzyskany obrys dłoni nie jest idealny, palce wskazujący, środkowy i serdeczny zostały nieco skrócone, stracono również część dłoni pod kciukiem. Jednak w porównaniu z przedstawionym przykładem w tym przypadku stracono dużo mniej z pierwotnego obrazu.

Skrypt w całości:

```
all = imread('ruch1.jpg');
tlo = imread('ruch0.jpg');
odj1= imsubtract(all,tlo);
odj2= imsubtract(tlo,all);
imwrite(odj1, 'odjęcie1.jpg');
imwrite(odj2, 'odjęcie2.jpg');
reka = imadd(odj1, odj2);
imwrite(reka, 'etap1-reka.jpg');
binarny = im2bw(reka, 0.05);
filtrowany= medfilt2(binarny, [3 3]);
imwrite(binarny, 'etap2-binarny.jpg');
imwrite(filtrowany, 'etap3-szumy.jpg');
se = strel('square',2);
proba3 = medfilt2(imerode(filtrowany, se), [3 3]);
imwrite(proba3, 'etap4-morfologia.jpg');
wypelnianie = imfill(proba3, 'holes');
imwrite(wypelnianie, 'etap5-wypelnianie.jpg');
krawedz1 = edge(wypelnianie, 'sobel');
krawedz2 = edge(wypelnianie, 'canny');
imwrite(krawedz1, 'etap6b-krawedziesobela.jpg');
imwrite(krawedz2, 'etap6-krawedziecannego.jpg');
krawedz = uint8(krawedz2)*255;
reszta = rgb2gray(all);
ostateczny = imadd(krawedz, reszta);
imwrite(ostateczny, 'ostateczny.jpg');
```