

Wydział Mechaniczny Politechniki Białostockiej

SPRAWOZDANIE Z PRZEDMIOTU

Systemy sterowania robotów

Kod przedmiotu: **MYAR2S01005M**

Temat: Barwa w obrazach cyfrowych.

Imię i nazwisko: Janusz Chmaruk, Jakub Dacewicz

Kierunek studiów: Automatyka i Robotyka

Specjalność: -

Semestr: I

Rok akademicki: 2022/2023

Data wykonania pracy: 28.05.2023

.....
podpis studenta

Weryfikacja efektów kształcenia:

EK1

EK5

EK2

EK6

EK3

EK7

EK4

EK8

Uwagi prowadzącego:

Ocena sumaryczna:

.....

podpis prowadzącego

Spis treści

1 Cel ćwiczenia	2
2 Kod	2
3 Wyniki	6
4 Wnioski	8

1 Cel ćwiczenia

Celem ćwiczenia jest przybliżenie możliwości wykorzystywania biblioteki Image Processing Toolbox (IPT) do przetwarzania i analizy informacji o barwie zawartej w obrazie cyfrowym. Zakres laboratorium obejmuje: konwertowanie obrazów cyfrowych do innej przestrzeni barw (opisu w różnych modelach barwnych), pseudokolorowanie, praktyczne wykorzystanie wybranych metod i technik analizy barwnej obrazów cyfrowych.

2 Kod

```
1 % Wyczyszczenie środowiska
2 clc
3 clear all
4 close all
5
6 % Wczytanie obrazu
7 [obrazRGB] = imread('picture1.jpg');
8
9 % Wydzielenie składowych R, G, B
10 obrazR = obrazRGB(:,:,1);
11 obrazG = obrazRGB(:,:,2);
12 obrazB = obrazRGB(:,:,3);
13
14 % Pobranie rozmiarów obrazu
15 [w, k, ~] = size(obrazRGB);
16
17 % Wyświetlenie oryginalnego obrazu
18 figure()
19 imshow(obrazRGB)
20 title('Obraz RGB')
21
22 % Utworzenie palet kolorów dla
23 % składowych R, G, B
24 r=0:1:255;
25 rr=r';
26 z=zeros(256,3);
27 z(:,1)=rr;
28 red=z/255;
29
30 r2=0:1:255;
31 rr2=r2';
32 z2=zeros(256,3);
33 z2(:,2)=rr2;
34 green=z2/255;
35
36 r3=0:1:255;
37 rr3=r3';
38
39 z3=zeros(256,3);
40 z3(:,3)=rr3;
41 blue=z3/255;
42
43 % Wartość nasycenia do osiągnięcia (Z)
44 Z = 90;
45
46 % Zmniejszenie nasycenia dla
47 % poszczególnych kanałów barwy
48 obrazR_redukowany = obrazR * (Z/255);
49 obrazG_redukowany = obrazG * (Z/255);
50 obrazB_redukowany = obrazB * (Z/255);
51
52 % Wygenerowanie wykresów 3D dla
53 % składowych kolorów
54 figure()
55 mesh(1:k, w:-1:1, obrazR_redukowany,
56 'FaceColor', 'texturemap', 'EdgeColor',
57 'none', 'CdataMapping', 'direct');
58 colormap(red)
59 view([-15 70]);
60 colorbar('vert');
61 title('Wykres przestrzenny - kanał
62 czerwony (zredukowany)');
63 xlabel('Kolumna'); ylabel('Wiersz');
64 zlabel('Natężenie');
65
66 figure()
67 mesh(1:k, w:-1:1, obrazG_redukowany,
68 'FaceColor', 'texturemap', 'EdgeColor',
69 'none', 'CdataMapping', 'direct');
70 colormap(green)
71 view([-15 70]);
72 colorbar('vert');
73 title('Wykres przestrzenny - kanał
74 zielony (zredukowany)');
75 xlabel('Kolumna'); ylabel('Wiersz');
76 zlabel('Natężenie');
77
78 %% 2. Dla jednego, wybranego
79 % obrazu RGB dokonać konwersji do 3
80 % wybranych innych
81 % reprezentacji barwnych. Wyniki w
82 % postaci obrazów umieścić w sprawozdaniu.
83 Scharakteryzować
84 % dla każdego modelu barwnego rodzaj
85 % pozyskiwanej informacji barwnej.
86
87 % Konwersja na przestrzeń barwną HSV
```

```

79 obrazHSV = rgb2HSV(obrazRGB);
80
81 % Konwersja na przestrzeń barwną Lab
82 obrazLab = rgb2lab(obrazRGB);
83
84 % Konwersja na przestrzeń barwną YCbCr
85 obrazYCbCr = rgb2ycbcr(obrazRGB);
86
87 % Wyświetlanie obrazów w trzech różnych
88 % reprezentacjach barwnych
89 figure()
90 imshow(obrazRGB)
91 title('Obraz RGB')
92
93 figure()
94 imshow(obrazHSV)
95 title('Obraz w przestrzeni barwnej HSV')
96 xlabel('Odcień'); ylabel('Nasyście');
97 zlabel('Wartość')
98
99 figure()
100 imshow(obrazLab)
101 title('Obraz w przestrzeni barwnej Lab')
102 xlabel('Jasność'); ylabel('a');
103 zlabel('b')
104
105 figure()
106 imshow(obrazYCbCr)
107 title('Obraz w przestrzeni barwnej
108 YCbCr')
109 xlabel('Luminancja');
110 ylabel('Chrominancja Cb');
111 zlabel('Chrominancja Cr')
112
113 %% 3. Dla dowolnego obraz
114 %a) dokonać konwersji do obrazu w
115 %odcieniach szarości (rgb2gray)
116
117 % Konwersja na obraz w odcieniach
118 %szarości
119 obrazGray = rgb2gray(obrazRGB);
120
121 % Wyświetlanie obrazów
122 figure()
123 imshow(obrazRGB)
124 title('Obraz barwny')
125
126 figure()
127 imshow(obrazGray)
128 title('Obraz w odcieniach szarości')
129
130 %% Wybór obszaru zainteresowania
131 h = imrect; % Narzędzie do wyboru
132 %obszaru prostokątnego
133 position = wait(h);
134 x = round(position(1));
135 y = round(position(2));
136 width = round(position(3));
137 height = round(position(4));
138
139 % Wyodrębnienie obszaru zainteresowania
140 obszarZainteresowania =
141 obrazRGB(y:y+height-1, x:x+width-1, :);
142
143 % Obliczenie średnich wartości dla
144 %każdego kanału barwnego
145 sredniaR =
146 mean(mean(obszarZainteresowania(:,:,1)));
147 sredniaG =
148 mean(mean(obszarZainteresowania(:,:,2)));
149 sredniaB =
150 mean(mean(obszarZainteresowania(:,:,3)));
151
152 % Wyświetlanie profilu barwnego RGB
153 figure()
154 plot([1, 2, 3], [sredniaR, sredniaG,
155 sredniaB], 'o-', 'LineWidth', 2)
156 title('Profil barwny RGB dla wybranego
157 obiektu')
158 xlabel('Kanał barwny')
159 ylabel('Średnia wartość piksela')
160 set(gca, 'XTick', [1, 2, 3],
161 'XTickLabel', {'R', 'G', 'B'})
162 grid on
163
164 %% c) zmodyfikować wybrany kanał
165 %barwny (np. poprzez odjęcie, dodanie,
166 %mnożenie itd.
167 % elementów w macierzy), a następnie za
168 %pomocą funkcji cat scalić macierze;
169 %wyświetlić obraz;
170
171 %% Wybrany kanał barwny (np. czerwony - R)
172 kanal = obrazRGB(:,:,1);
173
174 % Modyfikacja kanału barwnego (np.
175 %dodanie wartości)
176 modyfikacja = kanal + 150;
177
178 % Scalenie macierzy
179 obrazModyfikowany = cat(3, modyfikacja,
180 obrazRGB(:,:,2), obrazRGB(:,:,3));
181
182 % Wyświetlanie obrazu
183 figure()
184 imshow(obrazRGB)
185 title('Obraz oryginalny')
186
187 figure()
188 imshow(obrazModyfikowany)
189 title('Obraz po modyfikacji')

```

```

174
175    %% d)      dokonać pseudokolorowania
176    %% (2 przypadki) dla 2 wartości: 64, 255
177    %% kolorów;
178
179    % Konwersja obrazu do obrazu w
180    % odcieniach szarości
181    obrazSzary = rgb2gray(obrazRGB);
182
183    % Pseudokolorowanie dla 64 kolorów
184    mapaKolorow64 = parula(64); % Wybór mapy
185    % kolorów (np. parula)
186    obrazPseudokolor64 = ind2rgb(obrazSzary,
187    mapaKolorow64);
188
189    % Pseudokolorowanie dla 255 kolorów
190    mapaKolorow255 = parula(255); % Wybór
191    % mapy kolorów (np. parula)
192    obrazPseudokolor255 =
193    ind2rgb(obrazSzary, mapaKolorow255);
194
195    % Wyświetlanie obrazów
196    figure()
197    subplot(2, 3, 1)
198    imshow(obrazSzary)
199    title('Obraz w odcieniach szarości')
200
201    subplot(2, 3, 2)
202    imshow(obrazPseudokolor64)
203    title('Pseudokolorowanie - 64 kolory
204    (parula)')
205
206    subplot(2, 3, 3)
207    imshow(obrazPseudokolor255)
208    title('Pseudokolorowanie - 255 kolorów
209    (parula)')
210
211    subplot(2, 3, 4)
212    imshow(obrazPseudokolor64)
213    title('Pseudokolorowanie - 64 kolory
214    (jet)')
215
216    subplot(2, 3, 5)
217    imshow(obrazPseudokolor255)
218    title('Pseudokolorowanie - 255 kolorów
219    (jet)')
220
221    %% e)      * wyodrębnić na obrazie
222    %% wybrany kolor.
223
224    % Konwersja obrazu do obrazu w
225    % odcieniach szarości
226    obrazSzary = rgb2gray(obrazRGB);
227
228    % Wyodrębnienie koloru czerwonego
229    obrazWyodrebniony = obrazRGB;
230    obrazWyodrebniony(repmat(~maskaCzerwony,[1,1,3])) =
231    0;
232
233    % Wyświetlanie obrazów
234    figure()
235    subplot(1, 2, 1)
236    imshow(obrazRGB)
237    title('Obraz oryginalny')
238
239    subplot(1, 2, 2)
240    imshow(obrazWyodrebniony)
241    title('Wyodrębniony kolor czerwony')
242
243    %% 4.      *Zmodyfikować kontrast
244    %% dowolnego obrazu (funkcja imadjust).
245    %% Sprawdzić różne wartości parametrów
246    %% tej funkcji. Dla wybranego przypadku
247    %% porównać
248    %% histogramy barw przed korekcją i po
249    %% korekcji gamma.
250
251    % Konwersja obrazu do obrazu w
252    % odcieniach szarości
253    obrazSzary = rgb2gray(obrazRGB);
254
255    % Wyświetlenie obrazu przed regulacją
256    % kontrastu
257    figure()
258    subplot(1, 3, 1)
259    imshow(obrazSzary)
260    title('Obraz przed regulacją kontrastu')
261
262    % Regulacja kontrastu dla różnych
263    % wartości parametrów
264    low_in = 0.2; % Dolne ograniczenie dla
265    % wartości wejściowych
266    high_in = 0.8; % Górnne ograniczenie dla
267    % wartości wejściowych
268    low_out = 0; % Dolne ograniczenie dla
269    % wartości wyjściowych
270    high_out = 1; % Górnne ograniczenie dla
271    % wartości wyjściowych
272
273    obrazSkorygowany1 = imadjust(obrazSzary,
274    [low_in high_in], [low_out high_out]);
275
276    low_in = 0.4; % Dolne ograniczenie dla
277    % wartości wejściowych

```

```

221    progG = 0;    % Dolny próg dla składowej
222    G
223
224    progB = 0;    % Dolny próg dla składowej
225    B
226
227    % Utworzenie maski dla koloru czerwonego
228    maskaCzerwony = (obrazRGB(:,:,1) >
229    progR) & (obrazRGB(:,:,2) > progG) &
230    (obrazRGB(:,:,3) > progB);
231
232    % Wyodrębniение koloru czerwonego na
233    % obrazie
234    obrazWyodrebniony = obrazRGB;
235    obrazWyodrebniony(repmat(~maskaCzerwony,[1,1,3])) =
236    0;
237
238    % Wyświetlanie obrazów
239    figure()
240    subplot(1, 2, 1)
241    imshow(obrazRGB)
242    title('Obraz oryginalny')
243
244    subplot(1, 2, 2)
245    imshow(obrazWyodrebniony)
246    title('Wyodrębniony kolor czerwony')
247
248    %% 4.      *Zmodyfikować kontrast
249    %% dowolnego obrazu (funkcja imadjust).
250    %% Sprawdzić różne wartości parametrów
251    %% tej funkcji. Dla wybranego przypadku
252    %% porównać
253    %% histogramy barw przed korekcją i po
254    %% korekcji gamma.
255
256    % Konwersja obrazu do obrazu w
257    % odcieniach szarości
258    obrazSzary = rgb2gray(obrazRGB);
259
260    % Wyświetlenie obrazu przed regulacją
261    % kontrastu
262    figure()
263    subplot(1, 3, 1)
264    imshow(obrazSzary)
265    title('Obraz przed regulacją kontrastu')
266
267    % Regulacja kontrastu dla różnych
268    % wartości parametrów
269    low_in = 0.2; % Dolne ograniczenie dla
270    % wartości wejściowych
271    high_in = 0.8; % Górnne ograniczenie dla
272    % wartości wejściowych
273    low_out = 0; % Dolne ograniczenie dla
274    % wartości wyjściowych
275    high_out = 1; % Górnne ograniczenie dla
276    % wartości wyjściowych
277
278    obrazSkorygowany1 = imadjust(obrazSzary,
279    [low_in high_in], [low_out high_out]);
280
281    low_in = 0.4; % Dolne ograniczenie dla
282    % wartości wejściowych

```

```

263 high_in = 0.6; % Górnne ograniczenie dla
264 % wartości wejściowych
265
266 obrazSkorygowany2 = imadjust(obrazSzary,
267 [low_in high_in], [low_out high_out]);
268
269 % Wyświetlenie obrazów po regulacji
270 % kontrastu
271 subplot(1, 3, 2)
272 imshow(obrazSkorygowany1)
273 title('Obrazy po regulacji kontrastu1')
274
275 subplot(1, 3, 3)
276 imshow(obrazSkorygowany2)
277 title('Obrazy po regulacji kontrastu2')
278
279 % Wyświetlenie histogramu przed korekcją
280 figure()
281 subplot(1, 2, 1)
282 imhist(obrazSzary)
283 title('Histogram przed korekcją')
284
285 % Korekcja gamma
286 gamma = 0.5; % Wartość gamma (dostosuj
287 % wg potrzeb)
288 obrazSkorygowany = imadjust(obrazSzary,
289 [], [], gamma);
290
291 % Wyświetlenie histogramu po korekcji
292 subplot(1, 2, 2)
293 imhist(obrazSkorygowany)
294 title('Histogram po korekcji gamma')
295
296 % Wyświetlenie obrazów
297 figure()
298 subplot(1, 2, 1)
299 imshow(obrazSzary)
300 title('Obraz przed korekcją')
301
302 subplot(1, 2, 2)
303 imshow(obrazSkorygowany)
304 title('Obraz po korekcji gamma')
305
306 %% 5.      **Dokonać pomiaru różnicy
307 % kolorów z wykorzystaniem funkcji deltaE
308 % oraz imcolordiff (można wykorzystać
309 % obraz lub jego fragment z
310 % punktu 3 oraz jego wersję
311 % zmodyfikowaną 3c).
312
313 % Obliczenie różnicy kolorów między
314 % obrazami
315 color_diff = imabsdiff(obrazRGB,
316 obrazModyfikowany);
317 color_diff2 = deltaE(obrazRGB,
318 obrazModyfikowany);
319
320 % Wyświetlenie różnicy kolorów
321 figure()
322 subplot(1, 2, 1)

```

```

313 imshow(color_diff);
314 title('Różnica kolorów (imabsdiff)');
315
316 subplot(1, 2, 2)
317 imshow(color_diff2);
318 title('Różnica kolorów (deltaE)');

```

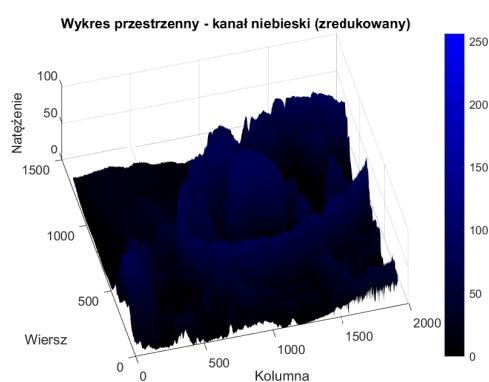
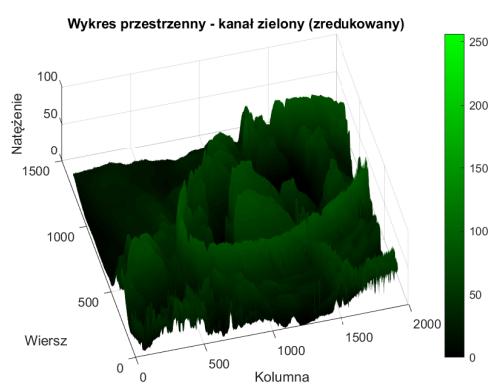
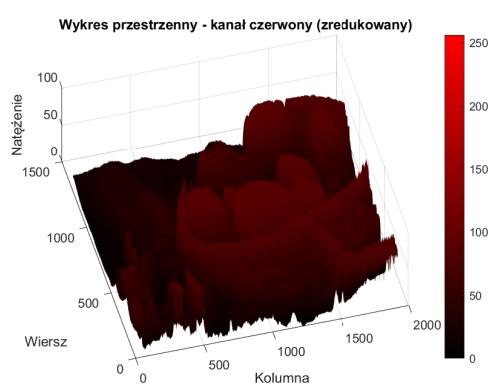
Pierwsza część kodu dotyczy wyświetlania rozkładu składowych barw RGB we współrzędnych 3D dla obrazu dog.jpg. Początkowo, kod wczytuje obraz i wyodrębnia składowe czerwony (R), zielony (G) i niebieski (B) do osobnych zmiennych. Następnie, tworzy mapy kolorów dla każdej składowej barwy. Wartość nasycenia Z jest następnie używana do redukcji nasycenia dla każdej składowej barwy, co skutkuje obrazami z zredukowanym nasyceniem dla składowych R, G, i B. Te obrazy są następnie wyświetlane jako wykresy przestrzenne.

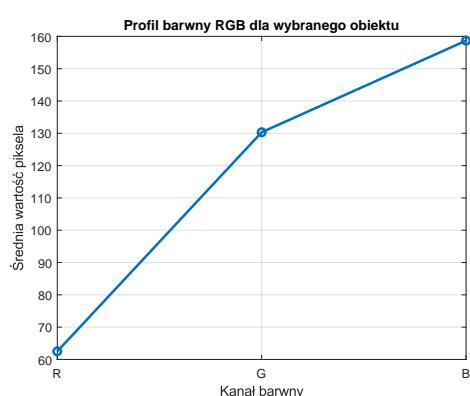
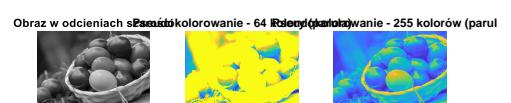
W drugiej części kodu, obraz RGB jest konwertowany do trzech różnych reprezentacji barwnych: HSV, Lab, i YCbCr. Te konwersje są użyteczne, ponieważ różne modele barwne mogą lepiej reprezentować pewne aspekty obrazu, takie jak percepcyjna jasność, nasycenie lub odcień. Wynikowe obrazy są następnie wyświetlane.

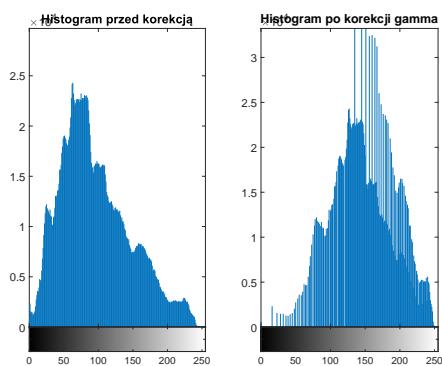
Trzecia część kodu obejmuje kilka operacji na obrazie barwnym. Obraz jest najpierw konwertowany do obrazu w odcieniach szarości. Następnie, jest tworzony profil barwny RGB dla wybranego fragmentu obrazu. Wybrany kanał barwny jest modyfikowany (w tym przypadku, poprzez dodanie wartości do kanału czerwonego), a następnie obrazy są skalowane z powrotem do obrazu RGB i wyświetlane. Kolejnym krokiem jest pseudokolorowanie obrazu dla dwóch różnych liczb kolorów (64 i 255), przy użyciu dwóch różnych map kolorów (parula i jet). Na końcu, z obrazu wyodrębniany jest wybrany kolor (w tym przypadku czerwony).

Czwarta część kodu skupia się na modyfikacji kontrastu obrazu. Zaczyna się od konwersji obrazu do obrazu w odcieniach szarości. Następnie, obraz jest wyświetlany przed regulacją kontrastu. Kontrast jest regulowany dla różnych wartości parametrów za pomocą funkcji imadjust. Na końcu, obrazy po regulacji kontrastu są wyświetlane.

3 Wyniki







jak wyświetlanie profilu barwnego RGB dla wybranego fragmentu obrazu oraz pomiar różnicy kolorów między obrazami.

4. **Korekcja kontrastu:** Kod przedstawia różne metody regulacji kontrastu obrazu, takie jak regulacja za pomocą funkcji imadjust oraz korekcja gamma. Porównanie histogramów przed i po korekcji kontrastu może pomóc w ocenie skuteczności tych technik.
5. **Użycie funkcji MATLABa:** Przedstawiony kod wykorzystuje różne funkcje dostępne w MATLABie do przetwarzania i analizy obrazów, takie jak imread, rgb2gray, ind2rgb, imadjust, imabsdiff itp. Zrozumienie i umiejętność korzystania z tych funkcji może być przydatne przy pracy z obrazami w MATLABie.

4 Wnioski

1. **Obróbka obrazów RGB w MATLABie:** Przedstawiony kod demonstruje różne techniki obróbki obrazów RGB, takie jak zmiana nasycenia, konwersja do innych przestrzeni barwnych, konwersja do odcieni szarości, modyfikacja kanałów barwnych, pseudokolorowanie oraz wyodrębnianie wybranych kolorów.
2. **Przestrzenie barwne:** Kod pokazuje, jak można skonwertować obrazy RGB do innych przestrzeni barwnych, takich jak HSV, Lab i YCbCr. Każda przestrzeń barwna dostarcza różne informacje na temat barwy, nasycenia i jasności.
3. **Analiza obrazów:** Przedstawiony kod pokazuje, jak przeprowadzić analizę obrazów, taką