

Systemy wizyjne

[Strona główna](#) / [Moje kursy](#) / [SW](#) / [13 marzec - 19 marzec](#)
/ [Podstawowe operacje przeprowadzane na obrazach cyfrowych.](#)

Podstawowe operacje przeprowadzane na obrazach cyfrowych.

Autorzy: Tomasz Kryjak, Piotr Pawlik

Tematyka:

- zapoznanie z podstawowymi operacjami przeprowadzanymi na obrazach cyfrowych:
- typu LUT (operacja jednoargumentowa)
- arytmetycznymi (operacje dwuargumentowe): dodawanie, odejmowanie, mnożenie, dzielenie
- logicznymi (operacje jedno i dwuargumentowe): AND, OR, XOR, NOT

Ćwiczenie:

Operacje LUT

Operacja LUT - polega na przekształcaniu wartości poszczególnych pikseli obrazu przy użyciu z góry przygotowanych tabel przekodowań (tabel korekcji).

1. Otwórz program **Matlab**. Ustal ścieżkę **Current Directory** na swój własny katalog w folderze **dodatkowe** Utwórz nowy m-plik.
2. Operację LUT realizuje się za pomocą funkcji `intlut`. Przy czym najważniejsze jest stworzenie odpowiedniej tablicy przekodowania. W przetwarzaniu obrazów najczęściej wykorzystuje się następujące funkcje:

- typu kwadratowa, pierwiastek kwadratowy
- typu logarytm, odwrócony logarytm

Nasze serwisy wykorzystują ciasteczka (cookies). Korzystając z nich wyrażasz zgodę na używanie cookies zgodnie z aktualnymi ustawieniami Twojej przeglądarki stron WWW. Rozumiem

3. Wczytaj przygotowany plik z przekodowaniami LUT (wcześniej ściągnij archiwum ze strony moodla i rozpakuj w "swoim" katalogu). Wykorzystaj polecenie: `load funkcjeLUT;`. Przekodowań jest siedem. Zostaną one wczytane do przestrzeni roboczej. Ich nazwy widoczne są w okienku **Workspace**.
4. Wyświetl przykładową funkcję. Wykorzystaj polecenie `plot (np.plot (kwadratowa);`); (Wcześniej dobrze jest zadeklarować wykres - `np.figure (1);`)
5. Wczytaj przykładowy obraz i wyświetl go - do wyboru "lena.bmp" lub "jet.bmp". Przypomnienie: wczytywanie `imread`, wyświetlanie `imshow`, kolejna `figure` - `figure (numer)`.
6. Na wybranym obrazie wykonaj operację LUT - na początek z tablicą przekodowań "kwadratowa". Wynik wyświetl. Podpowiedź: sprawdź w helpie jak działa funkcja `intlut`.
7. Aby lepiej zobaczyć w jaki sposób działają różne przekodowania LUT skonstruujemy funkcję, która jako argumenty pobierać będzie obrazek oryginalny oraz tablicę przekodowania, a następnie na wspólnym rysunku będzie wyświetlać: funkcję, obraz wejściowy oraz wynik przekodowania. (Przy okazji zobaczmy/przypomnimy sobie jak tworzy się funkcje w Matlabie oraz poznamy/przypomnimy/utrwalimy polecenie `subplot`).
 - utwórz nowy m-plik
 - w pierwszej linii umieść następujący kod: `function`
`LUT (obraz,przekodowanie) end.`
 - oczywiście kod należy umieścić pomiędzy nagłówkiem funkcji a słowem `end`
 - wywołaj przekodowanie LUT - tak jak w punkcie 6
 - wyświetl wyniki:
 - `subplot` powinien składać się z trzech pól - wykres przekodowania i dwa obrazy (oryginalny i przekształcony)
 - można zastosować układ 2x2 i górny wykres połączyć (`subplot (2,2,1:2)`) lub układ 1 x 3
 - każdy wykres powinien być podpisany (`title`)
 - aby wykres przekodowania wyglądał "porządnie" można wykorzystać następujące funkcje: `xlim`, `ylim`, `daspect`. Szczegóły w helpie.
 - zapisz m-plik, Matlab sam zaproponuje nazwę LUT.m
8. W "głównym" m-pliku wywołaj stworzoną funkcję. Najpierw utwórz *figure* odpowiednim numerem a następnie wywołaj funkcję LUT z odpowiednimi argumentami. Aby przejrzeć wyniki wszystkich przekodowań konieczne jest stworzenie siedmiu wykresów.

DODAWANIE

1. Utwórz nowy m-plik. Nazwij go i zapisz. Nie zapomnij o poleceniach `clear all;`
`close all;`. Wczytaj dwa obrazy 'lena.bmp' i 'jet.bmp' i wyświetl je.
2. Dodaj obrazy **Lena** i **Jet**, wykorzystaj funkcję `imadd` (sposób jej użycia należy sprawdzić w pomocy Matlaba). Uzyskany wynik wyświetl.
3. Czy wynik sumowania jest satysfakcjonujący? Co może niekorzystnie wpływać na rezultat operacji? Funkcja `imadd` ma możliwość podania typu danych w jakim może być zapisany wynik. Spróbuj wykorzystać typ **uint16**. Uwaga do poprawnego wyświetlania potrzebna jest następująca modyfikacja: `imshow(sum, []);`. Parametr `[]` oznacza, że dane z obrazu **sum** zostaną przed wyświetleniem przeskalowane do zakresu 0-255, przy czym jako 0 zostanie wzięte `min(sum)`, a jako 255 `max(sum)`.

KOMBINACJA LINIOWA

3. Do wykonywania operacji kombinacji liniowej służy funkcja `imlincomb`. Zapoznaj się z dokumentacją tej funkcji i przetestuj kilka kombinacji liniowych obrazów **Lena** i **Jet**.

ODEJMOWANIE

5. Wykorzystując funkcję `imsubtract` odejmij obrazy **Lena** i **Jet**.
6. Czy wynik odejmowania jest satysfakcjonujący ? Co może niekorzystnie wpływać na rezultat operacji? (Odpowiedz w komentarzu w m-pliku) Rozwiązaniem problemu jest zmiana typu danych dla obrazów **Lena** i **Jet** z **uint8** na **int16**. Odpowiedz na pytanie dlaczego zmiana typu poprawia wynik odejmowania? (Odpowiedz w komentarzu w m-pliku) Przydatna składnia: `lena16 = int16(lena);` Podczas wyświetlania pamiętaj o przeskalowaniu (`[]`).
7. Często zamiast zwykłego odejmowania wykorzystuje się operację wartość bezwzględna z różnicy (pozwala to m. in. uniknąć pokazanych powyżej problemów). Wykorzystując funkcję `imabsdiff` wykonaj operację wartość bezwzględna z różnicy dla obrazów **Lena** i **Jet**.

MNOŻENIE

7. Mnożenie dwóch obrazów pozwala wykonać funkcja `immultiply`. Wykonaj mnożenie obrazów **Lena** i **Jet** - czy wynik takiej operacji zawiera jakąś istotną informację ? Dlaczego? (Odpowiedz w komentarzu w m-pliku)
9. Mnożenie częściej wykorzystuje się jako
 - mnożenie przez stałą - co powoduje ogólne rozjaśnianie albo ściemnianie obrazu

Zamień wczytaną macierz na typ boolean (np. `maska = boolean(maska);`)
Przemnóż wybrany obraz przez maskę.

NEGATYW

10. Często wykorzystywaną operacją jest negatyw (pokazany wcześniej przy okazji operacji LUT) - funkcja `imcomplement`. Przetestuj jej działanie.
11. Przedstaw wyniki prowadzącemu.

Operacje logiczne:

Na poszczególnych punktach obrazu (najczęściej binarnego - czyli składającego się z dwóch kolorów: czarnego i białego) można wykonywać operacje logiczne: NOT, AND, OR, XOR itp.

1. Utwórz nowy m-plik. Nazwij go i zapisz. Nie zapomnij o poleceniach `clear all;` `close all;`. Wczytaj dwa obrazy 'kolo.bmp' i 'kwadrat.bmp'.
2. Zamień wczytane obrazy na typ boolean (np. `kolo = boolean(kolo);`). Wyświetl wczytane obrazy.
3. Na wczytanych obrazach wykonaj wybrane operacje logiczne. NOT (operator '~'), AND ('&'), OR ('|'), XOR (xor). Rezultaty wyświetl.
4. Przedstaw wyniki prowadzącemu..

Ostatnia modyfikacja: wtorek, 13 marzec 2018, 10:53



Platforma e-Learningowa obsługiwana jest przez:
Centrum e-Learningu AGH oraz Uczelniane Centrum Informatyki AGH

Nasze serwisy wykorzystują ciasteczka (cookies). Korzystając z nich wyrażasz zgodę na używanie cookies zgodnie z aktualnymi ustawieniami Twojej przeglądarki stron WWW. Rozumiem