

APO: Photoshop

Temat projektu:

Program p~~o~~i automatycznej konwersji sekwencji obraz~~o~~w w odcieniach szaro~~s~~ci na sekwencj~~e~~ obraz~~o~~w kolorowych.

Wykorzystanie wygenerowanych sekwencji obraz~~o~~w:

- w~~t~~asnych
- istniej~~a~~cych

Ko~~l~~o~~r~~a, tr~~o~~j~~k~~ą~~t~~y, kwadraty, intensywno~~s~~ć szaro~~s~~ci na intensywno~~s~~ć jasno~~s~~ci w HSV.

Jak uruchomić aplikację?

Z użyciem przeglądarki

- Ściągamy projekt na dysk komputera <https://github.com/piecioshka/apo-photoshop/archive/master.zip> (<https://github.com/piecioshka/apo-photoshop/archive/master.zip>).
- Rozpakowujemy paczkę z projektem apo-photoshop.
- Pobieramy (paczkę zip) node-webkit ze strony <https://github.com/rogerwang/node-webkit#downloads> (<https://github.com/rogerwang/node-webkit#downloads>) na nasz system operacyjny.
- Rozpakowujemy paczkę z projektem node-webkit.
- Kopiujemy zawartość projektu apo-photoshop to katalogu który powstał po rozpakowaniu node-webkit.
- Uruchamiamy plik nw.exe (Windows) lub nw (Linux) lub node-webkit.app (Mac).

Z użyciem konsoli (dla programistów)

```
$ git clone https://github.com/piecioshka/apo-photoshop.git
$ cd apo-photoshop
$ npm install
$ npm start
```

Wykorzystywane narzędzia

Silnik

- node.js - <http://nodejs.org/> (<http://nodejs.org/>)
- node-webkit - <https://github.com/rogerwang/node-webkit> (<https://github.com/rogerwang/node-webkit>)

Obróbka obrazów

- Canvas - <http://www.w3.org/TR/2dcontext/> (<http://www.w3.org/TR/2dcontext/>)

Inne pomocne narzędzia

- Gulp - <http://gulpjs.com/> (<http://gulpjs.com/>)
- Underscore.js - <http://underscorejs.org/> (<http://underscorejs.org/>)
- MoveMaster.js - <https://github.com/piecioshka/move-master.js> (<https://github.com/piecioshka/move-master.js>)
- promise.js - <https://github.com/stackp/promisejs> (<https://github.com/stackp/promisejs>)
- JSHint - <http://www.jshint.com/docs/options/> (<http://www.jshint.com/docs/options/>)

Wszystkie wykorzystywane narzędzia są darmowe.

Lista zadań

- ☒ Skróty klawiaturowe
- ☒ Słowniki tłumaczeń
- ☒ Drag & drop na oknach
- ☒ Wyświetlanie histogramu
- ☒ Ćw. 1
 - Zadanie 1
 - ☒ Metoda średnich
 - ☒ Metoda losowa
 - ☒ Metoda sąsiedztwa
 - ☒ Dodatkowo wymyślić jeden swój sposób
- ☐ Ćw. 2
 - Zadanie 1
 - ☒ Operacja odwrotności (negacji)
 - ☒ Operacja progowania (binaryzacji)
 - ☒ Operacja redukcji poziomów szarości
 - ☒ Operacja rozciągania
 - ☒ Regulacja jasnością
 - ☒ Regulacja kontrastem
 - ☒ Regulacja korekcją gamma
 - Zadanie 2
 - ☐ Uniwersalna operacja punktowa jednoargumentowa (oparta na tablicy LUT z możliwością zadawania parametrów w sposób interakcyjny (np. poprzez modyfikację postaci graficznej Uniwersalnego Operatora Punktowego)).
 - ☐ Tablica LUT
 - Zadanie 3
 - ☐ Typowe operacje punktowe dwu i wieloargumentowe (arytmetyczne (ADD, SUB,...) i logiczne (OR, AND, XOR,...)).
- ☐ Ćw. 3
 - Zadanie 1
 - ☐ Operacje wygładzania liniowego oparte na 4 typowych maskach wygładzania, wyostrażania liniowego oparte na 4 maskach laplasjanowych, detekcji krawędzi oparte na 3 maskach detekcji krawędzi.
 - ☐ Uniwersalna operacja liniowa (wygładzanie i wyostrażanie oparte na masce 3x3 o wartościach zadawanych w sposób interakcyjny). *Uwaga: zastosować opcjonalnie znane metody operacji na skrajnych wierszach i kolumnach obrazu oraz 3 metody skalowania (w przypadku operacji wyostrażania)*
 - Zadanie 2
 - ☐ Uniwersalna operacja medianowa (otoczenie 3x3, 3x5, 5x5, 7x7 itd.). *Uwaga: zastosować opcjonalnie znane metody operacji na skrajnych wierszach i kolumnach obrazu.*
 - Zadanie 3
 - ☐ Uniwersalna operacja logiczna wygładzania (kierunek 0, 1, 2, 3). *Uwaga: zastosować opcjonalnie wybrane metody operacji na skrajnych wierszach i kolumnach obrazu.*
 - Zadanie 4
 - ☐ Operacje wyostrażania gradientowego (2 maski uniwersalne, 2 maski Robertsa, 2 maski Sobela).

Uwaga: zastosować opcjonalnie wybrane metody operacji na skrajnych wierszach i kolumnach obrazu oraz 3 metody skalowania

- ☐ Ćw. 4

- Zadanie 1

- ☐ Operacja liniowa sąsiedztwa oparta na masce 5x5 utworzonej na podstawie dwóch masek 3x3 użytych w dwuetapowej (1-szy etap – wygładzanie, 2-gi etap – wyostrzanie) operacji filtracji. Opracowaną aplikację przetestować na wybranych obrazach i porównać wyniki otrzymane przy użyciu maski 5x5 z wynikami uzyskanymi przy użyciu kolejno dwóch masek 3x3. Uwaga: zastosować opcjonalnie 5 znanych z wykładu metod operacji na skrajnych wierszach i kolumnach obrazu oraz 3 znane metody skalowania (proporcjonalna, trójwartościowa, obcinająca).

- Zadanie 2

- ☐ Korzystając z podanego na wykładzie algorytmu ścieniania zrealizować program przekształcający utworzony obiekt, np. literę (lub 2 litery – np. inicjały wykonawcy) w szkielec (szkielecy).

- Zadanie 3

- ☐ Operacje erozji, dylatacji, otwarcia, zamknięcia dla dwóch przypadków elementu strukturalnego:
 - ☐ a) romb (cztero-sąsiedztwo)
 - ☐ b) kwadrat (ośmio-sąsiedztwo)

- ☐ Ćw. 5

- Zadanie 1

- ☐ Segmentacja obrazów z wykorzystaniem: progowania, rozrostu obszaru, dołączania, podziału, podziału i dołączania - algorytm i aplikacja.

- Zadanie 2

- ☐ Segmentacja oparta na opisie tekstury;
 - ☐ a) obliczanie deskryptorów tekstury (texture descriptors),
 - ☐ b) obliczanie histogramów różnic poziomów jasności (histograms of gray-level differences),
 - ☐ c) obliczanie ciągów pikseli o takiej samej wartości (run length statistics) - algorytm i aplikacja.

- Zadanie 3

- ☐ Segmentacja oparta na opisie tekstury;
 - ☐ a) obliczanie wartości prawdopodobieństwa pojawienia się pary pikseli o zadanych poziomach jasności w odległości d jeden od drugiego (obliczanie macierzy współwystąpień (co-occurrence matrix calculation)),
 - ☐ b) wyznaczenie rozkładu widma potęgowego (power spectrum) - algorytm i aplikacja.

- Zadanie 4

- ☐ Opis obrazu z wykorzystaniem algorytmu żółwia.

- ☒ Duplikacja aktywnego okna

- ☐ Resize okna

- ☒ Przywróć obrazek do pierwotnego stanu

- ☒ Zamknięcie programu

- ☒ Aktywacja właściwych elementów w menu kiedy aktywne jest odpowiednie okno.

- ☒ Konwertuj otwierane kolorowe obrazy do postaci obrazu w odcieniach szarości.

- ☒ Przesuwać oknem klikając w tytuł okna.

Zasada działania projektu

- ☒ Wczytanie sekwencji obrazów.
- ☐ Rozpoznanie obiektów z pierwszego kadru.
- ☐ Prezentacja wyodrębnionych obiektów w osobnym oknie.
- ☐ Dobór kolorów z palety HSV dla każdego obiektu.
- ☐ Zastosowanie koloru do pierwszego kadru.
- ☐ Algorytm iteracyjny:
 - ☐ Rozpoznanie obiektów na i-tym kadrze.
 - ☐ Dopasowanie go do rozpoznanego wcześniej obiektu.
 - ☐ Zastosowanie tego samego koloru co rozpoznany obiekt.
- ☐ Po zakończeniu informacja o statusie powodzenia obrazu.

Przydatne materiały

- http://pl.wikipedia.org/wiki/Lista_czarno-bia%C5%82ych_film%C3%B3w_poddanych_koloryzacji (http://pl.wikipedia.org/wiki/Lista_czarno-bia%C5%82ych_film%C3%B3w_poddanych_koloryzacji)
- <http://www.cs.huji.ac.il/~yweiss/Colorization/> (<http://www.cs.huji.ac.il/~yweiss/Colorization/>)
- <https://github.com/cmisenas/canny-edge-detection> (<https://github.com/cmisenas/canny-edge-detection>)
- <http://mbs98.republika.pl/projekty/ro/ro.html> (<http://mbs98.republika.pl/projekty/ro/ro.html>)