APO: Photoshop

Temat projektu:

```
Program półautomatycznej konwersji sekwencji obrazów w odcieniach szarości na sekwencję obrazów
kolorowych.

Wykorzystanie wygenerowanych sekwencji obrazów:
 a) własnych
 b) istniejących

Koła, trójkąty, kwadraty, intensywność szarości na intensywność jasności w HSV.
```

Jak uruchomić aplikację?

Z użyciem przeglądarki

1. Ściągamy projekt na dysk komputera https://github.com/piecioshka/apo-photoshop/archive/master.zip (https://github.com/piecioshka/apo-photoshop/archive/master.zip).
2. Rozpakowujemy paczkę z projektem `apo-photoshop`.
3. 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.
4. Rozpakowujemy paczkę z projektem `node-webkit`.
5. Kopiujemy zawartość projektu `apo-photoshop` to katalogu który powstał po rozpakowaniu `node-webkit`.
6. 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/)
- Underscore.assert.js - https://github.com/piecioshka/underscore.assert.js (https://github.com/piecioshka/underscore.assert.js)
- 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

Zadanie zlecone przez prowadzącego:

- [x] Ćw. 1
    - Zadanie 1
        - [x] Wyświetlanie histogramu
        - [x] Metoda średnich
        - [x] Metoda losowa
        - [x] Metoda sąsiedztwa
        - [x] Dodatkowo wymyślić jeden swój sposób
- [x] Ćw. 2
    - Zadanie 1
        - [x] Operacja odwrotności (negacji)
        - [x] Operacja progowania (binaryzacji)
        - [x] Operacja redukcji poziomów szarości
        - [x] Operacja rozciągania
        - [x] Regulacja jasnością
        - [x] Regulacja kontrastem
        - [x] Regulacja korekcją gamma
    - Zadanie 2
        - [x] Uniwersalna operacja punktowa jednoargumentowa (oparta na tablicy LUT z możliwością zadawania parametrów w sposób interakcyjny (np. poprzez modyfikację postaci graficznej Uniwersalnego Operatora Punktowego)).
    - Zadanie 3
        - [x] Typowe operacje punktowe dwu i wieloargumentowe (arytmetyczne (ADD, SUB, MUL) i logiczne (OR, AND, XOR)).
- [ ] Ćw. 3
    - Zadanie 1
        - [x] a)
            - [x] Operacje wygładzania liniowego oparte na 4 typowych maskach wygładzania.
            - [x] Operacje wyostrzania liniowego oparte na 4 maskach laplasjanowych.
            - [x] Detekcja krawędzi oparta na 3 maskach detekcji krawędzi.
        - [ ] b)
            - [ ] Uniwersalna operacja liniowa (wygładzanie i wyostrzanie 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 wyostrzania).
    - 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 wyostrzania 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 szkielet (szkielety).
  - 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 piksli o zadanych poziomach jasności w odległości d jeden od drugiego (obliczanie macierzy współwystąpień (co-occurence matrix calculation)),
      - [ ] b) wyznaczenie rozkładu widma potęgowego (power spectrum) - algorytm i aplikacja.
  - Zadanie 4
    - [ ] Opis obrazu z wykorzystaniem algorytmu żółwia.

Zadania zlecone przez developerów:

- [x] Skróty klawiaturowe.
- [x] Słowniki tłumaczeń.
- [x] Drag & drop na oknach.
- [x] Duplikacja aktywnego okna.

- [x] Przywróć obrazek do pierwotnego stanu.
- [x] Zamknięcie programu.
- [x] Aktywacja właściwych elementów w menu kiedy aktywne jest odpowiednie okno.
- [x] Konwertuj otwierane kolorowe obrazy do postaci obrazu w odcieniach szarości.
- [x] Przesuwać oknem klikając w tytuł okna.

Uwagi zgłoszone na ost. zajęciach:

- [x] Histogram obok obrazka.
- [x] Aktualizacja histogramu, kiedy uruchamiamy jakąś operację.
- [x] Filtracja liniowa - na sztywno elementy, aby przesuwanie okna z opcjami po za okno programu nie łamało.
- [x] Duplikacja aktualnej wersji obrazka (po ewentualnych modyfikacjach).
- [x] Histogram tylko dla ostatniego obrazka.
- [x] Operacje tylko dla ostatniego obrazka.
- [x] Zamknięcie otwartych okien z opcjami kiedy zamkniemy obraz na którym te operacje są uruchamiane.
- [x] Zapisanie obrazu (zapytać przy zamknięciu zmodyfikowanego obrazka).
- [ ] Resize okna.
- [ ] Pomoc na Windowsie nie działa.
- [ ] Lista kanałów szarości do modyfikacji (UOP).
- [ ] Wybieranie operacji arytmetycznych oraz logicznych na podstawie otwartych okien, a nie wybrania kilku obrazów.
- [ ] Wygładzanie: maska 1 (źle).
- [ ] Do zaliczenia (ptak morf, i ptak morf bin):
    - [ ] Z 4 ćw. zadania (morfologiczne).
    - [ ] Algorytm żółwia.
- [x] Po najechaniu na histogram, prezentować pod nim: nr kanały szarości i ilość wystąpień.

Zadania potrzebne do zrealizowania projektu grupowego:

- [x] Wczytanie sekwencji obrazów.
- [ ] Rozpoznanie obiektów z pierwszego kadru.
    - [ ] (Progowanie przedziałami) wytniemy obrazek w zadanym odcieniu szarości.
- [ ] 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)
- http://www.algorytm.org/przetwarzanie-obrazow/filtrowanie-obrazow.html (http://www.algorytm.org/przetwarzanie-obrazow/filtrowanie-obrazow.html)