

# APO: Photoshop

## Temat projektu:

Program p<sup>o</sup>1 automatycznej konwersji sekwencji obraz<sup>o</sup>w w odcieniach szaro<sup>ś</sup>ci na sekwencj<sup>e</sup> obraz<sup>o</sup>w kolorowych.

Wykorzystanie wygenerowanych sekwencji obraz<sup>o</sup>w:

- w<sup>l</sup>asnych
- istniej<sup>a</sup>cych

Ko<sup>l</sup>o<sup>ś</sup>ć, tr<sup>o</sup>j<sup>k</sup>ąt<sup>y</sup>, kwadraty, intensywno<sup>ś</sup>ć szaro<sup>ś</sup>ci na intensywno<sup>ś</sup>ć jasno<sup>ś</sup>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
$ ~/apo-photoshop/ # npm install
$ ~/apo-photoshop/ # npm start
```

Bądź też można wygenerować paczkę na wszystkie platformy:

```
$ ~/apo-photoshop/ # gulp build
```

## 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/pecioshka/underscore.assert.js> (<https://github.com/pecioshka/underscore.assert.js>)
- MoveMaster.js - <https://github.com/pecioshka/move-master.js> (<https://github.com/pecioshka/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 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 – wyostrażanie) 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 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.

### **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 ostatnich 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).
- [x] Po najechaniu na histogram, prezentować pod nim: nr kanały szarości i ilość wystąpień (oraz dodać paletę kolorów).
- [x] Lista kanałów szarości do modyfikacji (UOP).
- [x] Wybieranie operacji arytmetycznych oraz logicznych na podstawie otwartych okien, a nie wybrania kilku obrazów.
- [x] Wygładzanie: maska 1 (źle).
- [x] Do zaliczenia (ptak morf, i ptak morf bin):
  - [x] Z ćw. 4 - zadania (morfologiczne).
  - [x] Z ćw. 5 - algorytm żółwia.
- [x] Pomoc na Windowsie nie działa.

### **Uwagi zgłoszone na dodatkowych zajęciach:**

- Do ćwiczeń
  - [x] Dodać słowo 'Kopia' do nazwy obrazka gdy go duplikujemy.
  - [x] Button zamknij wszystko okna.
  - [x] Operacje morfologiczne bez limitu wykonania.
  - [x] Skrót do zwiększania/zmniejszania wartości.
  - [x] Filtracja liniowa: Metoda trójwartościowa - źle działa.
  - [x] Filtracja liniowa: Filtracja dolnoprzepustowa - bez tej prawej kolumny.
- Do projektu
  - [x] Kolorować w nowym oknie, aby nie modyfikować oryginału.
  - [x] Wysegmentować jakąś postać z dowolnego obrazka
  - [x] Dorobić jeszcze z 2 sekwencje, bardziej rozbudowane (więcej obiektów).

### **Problemy znalezione podczas testów:**

- [x] Problem z szerokimi obrazkami w filtracji liniowej (analizuje tylko kwadrat).
- [x] Przywrócenie do pierwotnego stanu nowo stworzonego okna (Operacje Arytmetyczne i Logiczne).
- [x] Aktualizacja tytułu okienka dopiero po załadowaniu obrazka (problem z długimi nazwami plików).
- [x] Rozszerzamy operacje morfologiczne dla MS Windows.
- [x] Kolorowanie jako operacja.
- [x] Algorytm Żółwia: wybór koloru który będzie użyty do zaznaczenia granicy obiektu.

### **Zadania potrzebne do zrealizowania projektu grupowego:**

- [x] Wczytanie sekwencji obrazów.
- [x] Rozpoznanie obiektów z pierwszego kadru wykorzystując Progowanie przedziałami. Wytniemy obrazek w zadanym odcieniu szarości.

- [x] Prezentacja wyodrębnionych obiektów w osobnym oknie.
- [x] Dobór kolorów z palety HSV dla każdego obiektu.
- [x] Algorytm iteracyjny: Zastosowanie nowego koloru do wyodrębnionego obiektu.
- [x] Po zakończeniu nakładania koloru pokazujemy w nowym oknie wynik pokolorowanej sekwencji obrazów.

### Uwagi z egzaminu

- [x] Dodać prowadzący i APO do pomocy.
- [x] Dodać liczbę ile jest obrazków otwartych na projekcie (w belce otwartego okna).
- [x] Zmienić nazwę w menu "Wybór obiektów na obrazie"
- [x] Dodać paletę kolorów HSV (Skorzystać z systemowego).
- [x] Labelki przycisków w sekwencji poklatkowej przenieść do Locali i spolszczyć.

### Bonusy

- [ ] Resize okna (Duży obrazek w małym oknie programu).
- [ ] Kiedy otworzymy obrazek, który nie będzie się mieścił z powodu kaskadowości, ustawiamy go w pkt (0, 0).
- [ ] Wyeliminować mruganie w pkt (0, 0) kiedy pokazuje się nowe okno.
- [ ] Edycja pokolorowanej sekwencji.
- [ ] Regulacja progów, regulacja kolorów.

### Przydatne materiały

- Online
  - [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://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)) (Ostatni odczyt: 2014/12/16 13:00)
  - <http://www.cs.huji.ac.il/~yweiss/Colorization/> (<http://www.cs.huji.ac.il/~yweiss/Colorization/>) (Ostatni odczyt: 2014/12/16 14:00)
  - <https://github.com/cmisenas/canny-edge-detection> (<https://github.com/cmisenas/canny-edge-detection>) (Ostatni odczyt: 2014/12/16 15:00)
  - <http://mbs98.republika.pl/projekty/ro/ro.html> (<http://mbs98.republika.pl/projekty/ro/ro.html>) (Ostatni odczyt: 2014/12/16 16:00)
  - <http://www.algorytm.org/przetwarzanie-obrazow/filtrowanie-obrazow.html> (<http://www.algorytm.org/przetwarzanie-obrazow/filtrowanie-obrazow.html>) (Ostatni odczyt: 2014/12/16 17:00)
  - <http://www.axiomx.com/posterize.htm> (<http://www.axiomx.com/posterize.htm>) (Ostatni odczyt: 2015/01/06 21:37)
- Książki
  - Io. Pitas: Digital image processing, algorithms and applications, John Wiley & Sons, March 2000