

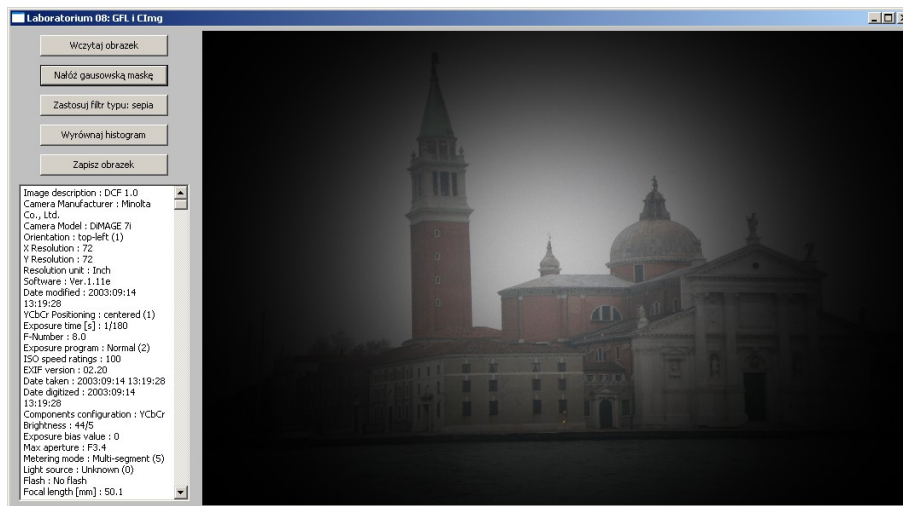
Biblioteka GFL i CImg.

Zadanie

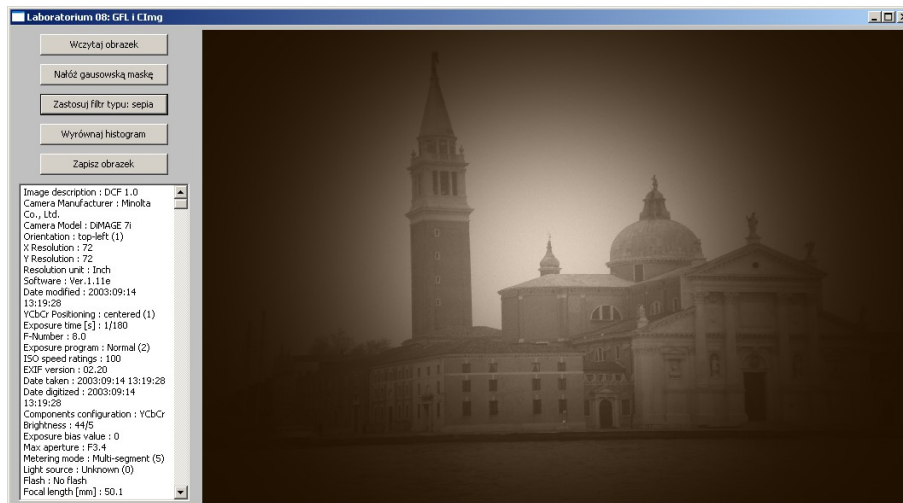
Napisać program wykonujący krok po kroku sekwencję przekształceń obrazu rastrowego z wykorzystaniem dwóch bibliotek omawianych na wykładzie. Minimalne wymagania wobec programu są następujące:



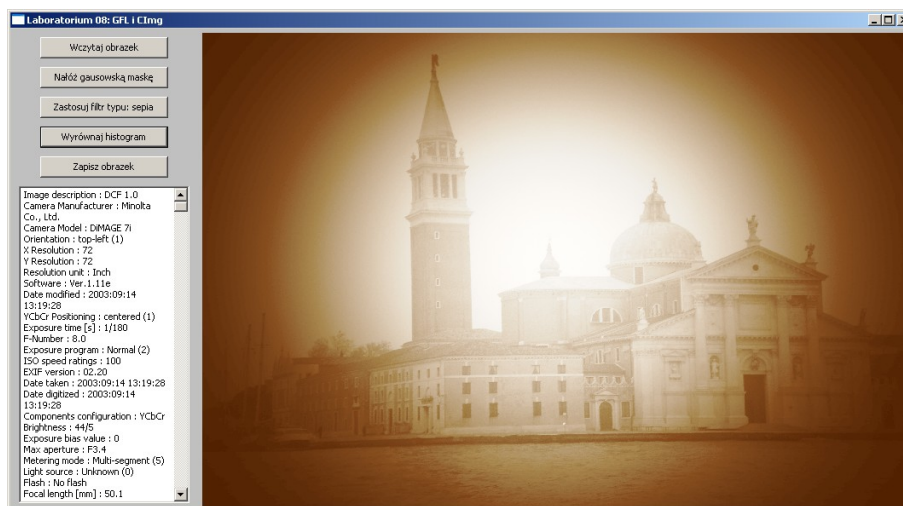
1. Po wciśnięciu przycisku program otwiera klasyczne okno dialogowe umożliwiające wybranie pliku do wczytania. Wczytywanie ograniczamy wyłącznie do plików w formacie *.jpg. Po wczytaniu obraz zostaje wyświetlony w głównym oknie programu. Przy zmianie rozmiarów okna obraz dostosowuje swoje rozmiary. W polu typu *Memo* po lewej stronie okna zostają wyświetlone wszystkie parametry zawarte w znacznikach EXIF.



2. Po wciśnięciu kolejnego przycisku na obraz zostaje nałożona maska ściemniająca obraz od centrum na zewnątrz zgodnie z funkcją Gaussa.



3. Użycie kolejnego przycisku nakłada na obraz filtr typu *sepia*.



4. Przedostatni przycisk wykonuje równoważenie histogramu.

5. Ostatni przycisk pozwala zapisać obraz do pliku w formacie *.png po wybraniu nazwy i katalogu docelowego w klasycznym oknie dialogowym.

Powyższy zestaw stanowi wymagane minimum. W ramach eksperymentów, pomysłowości i wolnego czasu można dodać dodatkowe przyciski generujące inne efekty.

Cel

Nabywanie praktycznej umiejętności przetwarzania obrazu z wykorzystaniem różnych bibliotek. Zapoznanie się ze sposobami łączenia bibliotek graficznych ze środowiskiem wxWidgets. Zrozumienie idei uogólnionych obrazów rastrowych o różnych typach pikseli (np. *float*) dostępnych w bibliotece Clmg.

Środki

Środowisko wxDev-C++, biblioteki GFL i Clmg.

Zarys możliwego rozwiązania

Okno i przyciski sterujące tworzymy w sposób typowy dla biblioteki wxWidgets i środowiska wxDev-C++, wzorując się na rozwiązaniach zaprezentowanych na poprzednich laboratoriach. Zarówno wxWidgets jak i każda z bibliotek pozwalają przechowywać w pamięci komputera obraz rastrowy, ale każde w innej postaci. Każda biblioteka używa własnego typu danych. Trzeba się zdecydować na jeden, w którym będziemy przechowywać wczytany obraz i tylko w razie potrzeby wykonywać konwersje do innych typów.

Najpierw należy wczytać obraz. Musimy to zrobić z wykorzystaniem biblioteki GFL, ponieważ tylko ona zapewnia obsługę meta danych typu EXIF. Po wczytaniu obrazu, w zależności od wcześniejszego wyboru, albo pozostawiamy obraz w zmiennej typu `GFL_BITMAP`, która jest natywnym sposobem przechowywania bitmap w bibliotece GFL albo konwertujemy do jednego z typów dostępnych w wxWidgets (np. `wxImage`) lub `CImg (CImg<unsigned char>)`.

W celu nałożenia maski ściemniającej należy skorzystać z możliwości biblioteki CImg. Biblioteka ta pozwala na stworzenie bitmapy o pikselach będących liczbami typu `float`. Na takiej bitmapie można zastosować funkcję dostępną w bibliotece wypełniającą obraz wartościami zmieniającymi się zgodnie z funkcją Gaussa. Biblioteka CImg udostępnia także zestaw funkcji wykonujących działania arytmetyczne na obrazach. Jeżeli teraz wczytany obraz pomnożymy przez obraz z wartościami `float`, to w miejscach o wartościach funkcji Gaussa mniejszych od jedynki obraz zostanie przyciemniony, a w tych o wyższych wartościach rozjaśniony.

Kolejny etap to zastosowanie filtru typu *sepia*. Filtr ten dostępny jest w bibliotece GFL.

Równoważenie histogramu można wykonać w dowolnej z obydwu bibliotek, ponieważ każda posiada odpowiednie funkcje. Należy jedynie pamiętać o konieczności przenoszenia obrazów pomiędzy zmiennymi różnych typów stosowanymi w różnych bibliotekach w zależności od wykonywanej operacji.

Na zakończenie, zapis do pliku najwygodniej będzie wykonać z wykorzystaniem tej biblioteki, w której przechowywano obraz.

Konwersję pomiędzy typami można zrealizować na wiele sposobów. Najprostsza metoda polega po prostu na odczycie kolorów punkt po punkcie z bitmapy zapisanej w zmiennej jednego typu i zapisywaniu punkt po punkcie do bitmapy reprezentowanej przez zmienną innego typu.

Jak się przygotować przed zajęciami

Źródła działającego programu liczą niewiele ponad 250 linii kodu, z czego ponad połowa tworzy interfejs użytkownika i zarządza nim. Wszystkie procedury konwersji obrazów i nakładania odpowiednich filtrów mieszczą się w granicach stu linii kodu. Wykonanie zadania wymaga jednak wcześniejszego zapoznania się z kilkoma zagadnieniami oraz przemyślenia strategii postępowania. Warto zatem:

- Zapoznać się ze źródłami programów demonstracyjnych prezentowanych na wykładach poświęconych bibliotekom GFL i CImg. Dotyczy to zwłaszcza programu do wczytywania danych EXIF i IPTC.

- Zapoznać się z dokumentacją obu bibliotek.
- Odnaleźć zmienne w jakich przechowywane są bitmapy w wxWidgets, GFL i CImg. W przypadku wxWidgets sugeruję użycie raczej **wxImage** niż **wxBitmap**.
- Odnaleźć funkcje służące do zapisu i odczytu poszczególnych pikseli w każdej z bibliotek.
- Przemyśleć strategię przechowywania obrazu i konwersji pomiędzy poszczególnymi bibliotekami.

Uwagi dotyczące kompilacji i konsolidacji

W celu skorzystania z obydwu bibliotek do programu należy dołączyć odpowiednie pliki nagłówkowe:

```
#include "libgfl.h"
#include "libgfle.h"
#include "CImg.h"
```

Należy również zadeklarować przestrzeń nazw:

```
using namespace cimg_library;
```

We właściwościach projektu należy dołączyć na etapie konsolidacji pliki: *libgfl.a* i *libgfle.a*. Pliki te albo muszą znajdować się w domyślnej ścieżce przeszukiwania albo należy je wskazać. W katalogu z plikiem wykonywalnym (albo w ścieżce), muszą się natomiast znaleźć pliki: *libgfl282.dll* i *libgfle282.dll*.