Propozycja II projektu zaliczeniowego z PiMN:

Obróbka obrazów (bitmap),

czyli elementy algebry liniowej

Opracowanie *Jędrzej Wardyn*

1 lipca 2024

Rysunek 1: Obraz lwa z ditheringiem (Bayer (ordered)) po lewej oraz oryginał po prawej

Wszelkie pytania proszę kierować na mail: `j.wardyn@uw.edu.pl`

1 Program do obróbki obrazów w formacie .bmp

Za czasów windowsa xp, visty, kiedy bywało, że nie było internetu ludzie wchodzili i rysowali sobie rzeczy w programie paint z nudów. Wejdźcie tutaj żeby poczuć się jak we wczesnych 2000cznych Taki program nie pozwalał na profesjonalną obróbkę obrazu, choć stanowił podstawę dla społeczności tworzących pixelarty, również gry w RPGmaker itp. Nas interesowałaby obróbka obrazów opierająca się na przekształceniach całego obrazu naraz o większej rozdzielczości niż 192x192 i być może niekoniecznie pixel po pixelu.

Chcielibyśmy mieć program, który mając dane obrazy będzie mógł dokonać na nich operacji typu sprowadzenie do skali szarości, obrót, inwersję, inwersję koloru i tak dalej, gdzie będzie można wybrać na jakim pliku działamy oraz operację i jakieś parametry (jeśli takie są) na przykład w konsoli:

```
./program nazwa_pliku.bmp obrot prawo 90,
```

gdzie ostatnia liczba reprezentuje kąt obrotu. Ale jeśli ktoś ma taki kaprys może też ująć zagadnienie interakcji z programem inaczej, niemniej tyle wystarczy.

Na zachętę macie tutaj do dyspozycji kod, który możecie modyfikować w celu utworzenia programu. Dana jest jedynie funkcja sprowadzenia do skali szarości. Przyjmuję tutaj format BMP z racji, że nie używają jakiejś skomplikowanej kompresji i nie trzeba się z tym mierzyć zbytnio. Wykorzystuje to otwartą bibliotekę na wszystkie platformy.

Niemniej korzystanie z tej biblioteki ani z C++ **NIE JEST** konieczne, ale jak ktoś chce na szybko to to wystarczy w zupełności.

Przykładowa funkcja w kodzie z linku powyżej zmieniająca obraz na odcienie szarości jest nieoptymalna: działa sekwencyjnie: pixel po pixelu. Proszę skorzystać z paralelizacji adekwatnie do przypadku Sprawdźcie użycie OpenMP.

Nie ograniczam tutaj jakie efekty można zastosować, niemniej nieco inspiracji mogę podać z kanału Acerola, konkretnie:
Sortowanie pixeli
Dithering
Filtr Kuwahary.

Tu jeszcze stronka do ditheringu: kliknij tu.

Obrazki do testów można ściągać z stronki z Royalty Free obrazkami. A konwertować dowolne obrazki (nie tylko png) do .bmp można na stronie do konwersji lub za pomocą programu do konwersji (wszystkie systemy: linux, mac, windows)

1.1 Co widziałbym, że ma się znaleźć:

1. Program może być odpalany z konsoli z argumentami funkcji main jaki plik ma przyjąć i co ma z nim robić. (Trochę jak ffmpeg). [OPCJONALNE]: Jak ktoś chce się pogimnastykować, to może spróbować zrobić coś na kształt menu do wyboru funkcji, coś bardziej graficznego, ale w konsoli to proszę bardzo.

2. Podstawa: program ma ALBO zapisywać output do plików które można otworzyć LUB/I bezpośrednio pokazywać rysunki. Istotna jest możliwość inspekcji tego, jak nasz program obrobi obraz.

3. W wysłanym rozwiązaniu chcę zobaczyć przykłady input/output od razu a nie tylko sam kod

4. Powinno być: obracanie obrazu, przynajmniej o 90stopni, nie wymagam utworzenia obracania o dowolny kąt, niemniej byłoby to również ciekawe(można potraktować wtedy jako opcję do wyboru).

5. Zmiana jasności obrazu.

6. Inwersja kolorów

do wyboru (ale wymagane) części wymienione na pierwszej stronie, prosiłbym o wykonanie dwóch lub więcej poniżej propozycje (możecie wymyślić własną funkcję też):

propozycja 1 Dithering, ale najlepiej kilka styli, Lista, (prawdopodobnie niekompletna) algorytmów, może być czarno-biały może być kolorowy dithering też, liczy się jakość wykonania.

propozycja 2 Anti-aliasing, gdzie również nieźle można się pobawić z opcjami, tutaj wyjaśnienie na temat anti-aliasingu

propozycja 3 Redukcja kolorów gdzie widziałbym dość szeroki wachlarz opcji, gdzie możemy wybrać paletę i to jak jakie kolory zmienia, co możemy wyrazić

1.2 Dla uproszczenia:

1. Nic nie musi dziać się w czasie rzeczywistym, program ma przyjąć, popracować sobie i oddać gotowy plik po przeróbkach.

2. Nie musimy tworzyć żadnego konwertera do .jpg lub .png, ani innych formatów, biblioteka jedynie ma zastosowanie do .bmp i innych bardzo prostych w strukturze plików. Ale nie zabraniam. [możliwość zgarnięcia punktów]

2 Wymagania i obwieszczenia:

Poniżej umieszczone są warunki, które wpływają na ostateczną punktację projektu.

Kod projektu:

1. Powinien posiadać czytelną i zrozumiałą instrukcję (w osobnym pliku) kompilacji i użytkowania programu.

2. **Powinien kompilować/uruchamiać się (brak tego warunku daje 0 punktów!).**
3. Powinien uruchamiać się bez błędów w stylu `segmentation fault`
4. Nie powinien skutkować jakimś szkodliwym działaniem, na przykład nadprodukcją zapisywanych plików zapełniających całą dostępną pamięć.
5. Przy wprowadzaniu lub użyciu danych, otwieraniu plików i innych czynnościach zależnych od pewnych czynników program powinien sprawdzać czy operacja została wykonana poprawnie, jeśli nie to powinien zostać pokazany odpowiedni do sytuacji komunikat błędu.
6. Kod programu powinien realizować wytyczne z opisu projektu.
7. (tam gdzie jest to możliwe) Warto jakby nie było zbytecznej redundancji
8. Warto by program nie był powolny lub nad wyraz zasobożerny polecam flagę (-O3 chyba, że macie wyraźne powody by jej nie użyć).
9. W kodzie dobrze jakby nie było części bezużytecznych (dodajcie `-Wall` do flag kompilacji), śmieci pozostałych po tworzeniu rozwiązania, proponuję wyczyścić kod a potem dodać komentarze.
10. W miarę możliwości kod ma być przejrzysty, opatrzony komentarzami w celu szybszego zrozumienia sprawdzającego, (co nie znaczy, że nie można używać jakichś trików). Komentarze wewnątrz pliku z kodem nie osobno!

Punkty 1-6 są najważniejsze, ale spełnienie ich nie gwarantuje maksymalnej ilości punktów.
Wszelkie pytania proszę kierować na mail: `j.wardyn@uw.edu.pl`