

Rozkład SVD i kompresja obrazów (10pkt)

TERMIN ODDANIA: 3.05.2017

DO ODDANIA: Raport w postaci pliku .pdf oraz projekt Qt , który powinien się uruchamiać w pracowni 411. W pliku `main.cpp` na samym początku powinna być zdefiniowana ścieżka do zdjęć, którą można łatwo zmieniać. Załącz też swoje obrazki, tak bym ewentualnie mógł na nich uruchomić Twój program.

UWAGA: Poniżej znajduje się opis co powinno znaleźć się w projekcie. Twój projekt powinien te wszystkie elementy zawierać, ale nie musi się tylko i wyłącznie do nich sprowadzać. W szczególności nie są tam wymienione wszystkie rzeczy krok po kroku, opis jest (zamierzenie) luźny.

Opis projektu

Wykonaj dekompozycje SVD dla obrazka

http://www.math.uni.wroc.pl/~lorek/image_analysis/images/baboon.bmp

oraz dla ok. 4 przez siebie wybranych czarno-białych obrazków (najlepiej tego samego rozmiaru bitmapy `.bmp`) oraz dwóch kolorowych (patrz dalsza część opisu). Postaraj się, aby były tam obrazy zarówno skomplikowane (prawie każdy mały obszar ma dużo szczegółów) jak i “proste” (np. jest dużo obszarów o jednakowym lub mało się zmieniającym kolorze)

Obrazy czarno-białe Każdy obraz f przedstaw jako

$$f = \sum_{i=1}^r \lambda_i^{\frac{1}{2}} \mathbf{u}_i \mathbf{v}_i$$

Nazwijmy $\lambda_i^{\frac{1}{2}} \mathbf{u}_i \mathbf{v}_i$ obrazem własnym (eigenimage) odpowiadającym wartości własnej λ_i . Posortujmy wartości własne w taki sposób, by $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r$. Oznaczmy $f_k =$

$\sum_{i=1}^k \lambda_i^{\frac{1}{2}} \mathbf{u}_i \mathbf{v}_i$. Dla każdego obrazka przedstaw (jako obrazek) kilka pierwszych obrazów

własnych oraz f_k dla kilku wybranych k (zwiększające się o stałą liczbę).

Przez kompresję obrazu f metodą rozkładu SVD nazywamy obraz f_k z odpowiednio dobranym parametrem k . Wypracuj jakiś sposób wyznaczania tego parametru (np. k ma być taką liczbą, by $\lambda_{k+1} + \dots + \lambda_r$ stanowiły maksymalnie jakiś ustalony procent całej sumy $\sum_{i=1}^r \lambda_i$, procent ten może być wybrany eksperymentalnie na jednym obrazku).

Następnie wylicz w tak ustalony sposób k dla każdego obrazka. Czy wielkość k zależy od stopnia skomplikowania obrazka?

Policz też ile pamięci zajmuje tak skompresowany obraz. Porównaj wynik do obrazów (standardowych) *.jpg* otrzymanych na chaosie za pomocą komendy:

```
chaos:~$ convert obrazek.bmp obrazek.jpg
```

Zakładamy, że wszystkie obrazy są 8-bitowe (= 1 bajt) (skala szarości 0-255). Zatem np. jeden wektor \mathbf{u}_i jest wektorem rozmiaru 256, i każdy element to 1 bajt, czyli \mathbf{u}_i jest zajmuje 256 bajtów = 0.25 Kb (1 Kb = 1024 bajty). Natomiast dla przykładu: obrazek 256×256 trzymany w postaci 8-bitowej bitmapy zajmuje: $256 \cdot 256 \cdot 8$ bitów = 524288 bitów = $524288 / (1024 \cdot 8) = 64$ Kb.

(UWAGA: rozmiar pliku *.bmp* może być większy z kilku powodów: np. mimo iż obraz jest czarnobiały, to każdy kanał r,g,b jest i tak pamiętany osobno, wtedy rozmiar to będzie $3 \cdot 64$ Kb, albo może być 16, 24 lub 32-bitowy, trzymane też mogą być informacje o rozdzielczości).

Obrazy kolorowe Dla każdego kanału R, G, B osobno wykonaj SVD i dokonaj kompresji. Potem osobno “odzyskaj” każdy kanał, przeskaluj do 0-255 i zaprezentuj odzyskany obraz kolorowy.