

Proszę wykonać oddzielnie podane niżej zadania dla obrazka „rzeczka.jpg”. Nie należy korzystać z gotowych funkcji dokonujących obróbki obrazka.



1. Proszę stworzyć 2 funkcje/skrypty porównujące obrazki (pokazujące różnice między obrazkami) i użyć ich do porównania obrazka oryginalnego i przetworzonego filtrem *gamma 1.5*.

Jako bazę niech posłużą 2 funkcję porównujące obrazki:

- **porównanie v.PS** – funkcja liczy moduł z różnicy między wartością nową a starą,

$$\text{wyjscie}(we_{stare}, we_{nowe}) = |we_{nowe} - we_{stare}|$$

Warto zwrócić uwagę, że jeśli obrazki będą jednakowe to powstanie czarny obraz.

- **porównanie v.2** – funkcja liczy różnicę z różnicy z wartości nowej a starą i wartością środkową (127.5)

$$\text{wyjscie}(we_{stare}, we_{nowe}) = \text{PrzytnijZakres}((we_{nowe} - we_{stare}) + \text{wartośćŚrodkowa})$$

gdzie *wartośćŚrodkowa* to 127.5 dla zakresu [0; 255] i 0,5 dla zakresu [0; 1]. Warto zwrócić uwagę, że jeśli obrazki będą jednakowe to powstanie szary obraz.

Zaimplementować należy rozszerzenie w/w funkcji, które będą wzmacniać widoczne różnice.

- **porównanie v.PS z mnożnikiem** – funkcja liczy iloczyn z mnożnika i modułu z różnicy między wartością nową a starą,

$$\text{wyjscie}(we_{stare}, we_{nowe}, \text{mnoznik}) = \text{PrzytnijZakres}(\text{mnoznik} * |we_{nowe} - we_{stare}|)$$

Warto zwrócić uwagę, że jeśli obrazki będą jednakowe to powstanie czarny obraz.

- **porównanie v.2 z mnożnikiem** – funkcja liczy różnicę z różnicy z wartości nowej a starą i wartością środkową (127.5)

$$\text{wyjscie}(we_{stare}, we_{nowe}, \text{mnoznik}) = \text{PrzytnijZakres}(\text{mnoznik} * (we_{nowe} - we_{stare}) + \text{wartośćŚrodkowa})$$

,

gdzie *wartośćŚrodkowa* to 127.5 dla zakresu [0; 255] i 0,5 dla zakresu [0; 1] . Warto zwrócić uwagę, że jeśli obrazki będą jednakowe to powstanie szary obraz.

org.



gamma 1.5



porównanie v.PS * 4



porównanie v.2 * 3



2. Istnieje wiele różnych sposobów zamiany obrazka na obrazek w odcieniach szarości, albo też po prostu wyciągnięcia z tego obrazka składowej opisującej jasność w przypadku przestrzeni YUV/Lab i podobnych. Dwoma najprostszymi metodami są podane niżej. Proszę ustalić wartość tych zmiennych dla całego obrazka i porównać je między sobą. Wartości te liczone są dla każdego piksela osobno na podstawie poszczególnych składowych kolorów tego piksela.

- **SzarośćŚrednia** – funkcja jako jasność poda wartość średnią ze składowej czerwonej, zielonej i niebieskiej.

$$\text{SzarośćŚrednia}(r, g, b) = \text{średnia}(r, g, b) \text{ ,}$$

gdzie r, g, b – to składowe kolorów danego piksela.

- **SzarośćSzary** – funkcja jako jasność poda sumę iloczynów odpowiednich mnożników przez składowe kolorów tego piksela. Wartość mnożników zależy od składowej i ich wartość jest podana poniżej.

$$\text{SzarośćSzary}(r, g, b) = 0,2126 * r + 0,7152 * g + 0,0722 * b$$

Szary



porównanie v.PS * 4



Srednia



porównanie v.2 * 3



3. Proszę wyliczyć składowe $Y'CbCr$ dla podanego obrazka, zgodnie ze wzorem podanym poniżej i wyświetlić ich wartości osobno. Wartości te liczone są dla każdego piksela oddzielnie na podstawie wartości odpowiednich kolorów mieszczących się w przedziale $[0; 255]$.

$$Y = 0 + 0.299 * R + 0.587 * G + 0.114 * B;$$

$$Cb = 128 - 0.168736 * R - 0.331264 * G + 0.5 * B; ,$$

$$Cr = 128 + 0.5 * R - 0.418688 * G - 0.081312 * B;$$

gdzie R, G, B to wartości składowych poszczególnych kolorów.

Ciekawostka - To, albo bardzo podobne przekształcenie, jest używane przy kompresji do formatu JPEG. Warto zauważyć, że składowa Y' określa jasność, a Cb i Cr opisują kolor danego piksela i są dość niezależne względem jasności tego piksela. Wszystkie te wartości mieszczą się w przedziale takim samym jak składowe kolorów R, G, B , czyli $[0; 255]$. Im

piksel jest jaśniejszy, tym wartości Y' jest wyższa. Wraz ze wzrostem nasycenia koloru wartości Cb , Cr bardziej oddalają się od 128. Wartość Cb można interpretować jako stopień dominacji koloru niebieskiego nad innymi, analogicznie Cr można czytać jako stopień dominacji koloru czerwonego nad innymi.

RGB



Y



Cb



Cr



4. Proszę dokonać konwersji obrazka na przestrzeń $Y'CbCr$, a następnie wrócić do RGB i porównać ten obrazek z oryginalnym. Wzory służące do zamiany są podane poniżej.

$$R = Y + 1.402 * (Cr - 128);$$

$$G = Y - 0.344136 * (Cb - 128) - 0.714136 * (Cr - 128);$$

$$B = Y + 1.772 * (Cb - 128);$$

porównanie v.PS * 1 000 000



porównanie v.2 * 1 000 000



Maksymalny moduł różnicy między obrazkiem oryginalnym, a przetworzonym na Y'CbCr i z powrotem powinien być mniejszy niż 0,0001.

5. Proszę dokonać konwersji obrazka na przestrzeń Y'CbCr, następnie dokonać zaokrąglenia dla tych wartości (Y, Cb, Cr), później przekształcić na RGB oraz ponownie dokonać zaokrąglenia (dla nowych R, G, B). Po skończonym procesie proszę porównać powstały obrazek z oryginalnym.

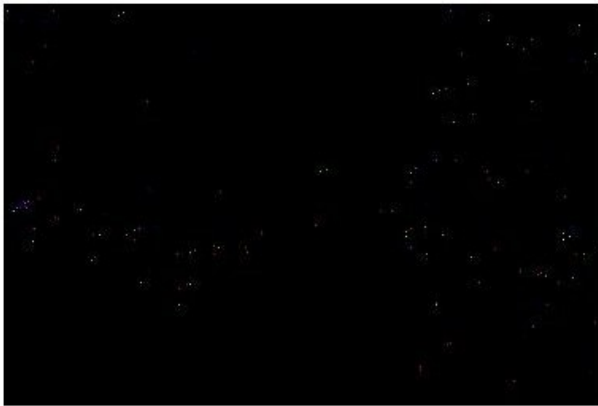
RGB org



to YCbCr, to RGB



porównanie v.PS * 250



porównanie v.2 * 300



Maksymalny moduł różnicy między obrazkiem oryginalnym, a przetworzonym powinien być równy 1.

6. Wewnątrz przestrzeni Y'CbCr to składowa Y zawiera większość informacji dot. jasności obrazka. Proszę stworzyć dwa obrazki, pierwszy nich będzie przetworzony filtrem *gamma 1.5* (wszystkie składowe R, G, B). Drugi obrazek należy zmienić na przestrzeń na Y'CbCr, obrobić tylko składową Y filtrem *gamma 1.5*, następnie zamienić na RGB. Proszę porównać oba obrazki.

RGB gamma 1.5



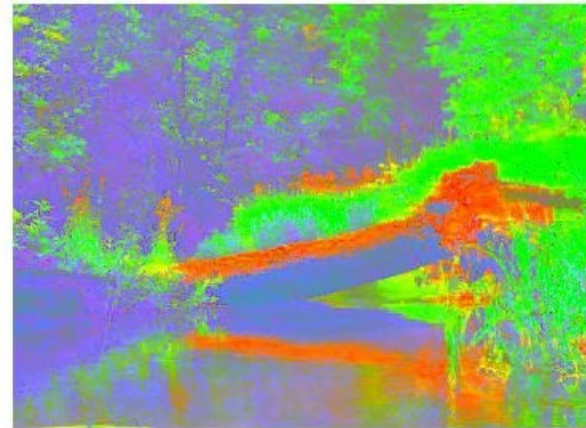
YCbCr, Y gamma 1.5, RGB



porównanie v.PS * 50



porównanie v.2 * 50



7. Wewnątrz przestrzeni $Y'CbCr$ to składowe Cb i Cr zawierają większość informacji dot. kolorów (a nie jasności) wewnątrz obrazka. Im wartość Cb i Cr jest bardziej odległa od 128 tym mamy do czynienia z bardziej nasyconym kolorem. Proszę zwiększyć nasycenie kolorów poprzez odsunięcie wartości Cb i Cr 1.5-krotnie od 128, zgodnie ze wzorem widocznym poniżej.

$$wyjscie = PrzytnijZakres(128 + (wejście - 128) * PodbicieKolorow) ,$$

gdzie *PodbicieKolorów* to mnożnik mający wpływ na zmianę kolorystyki, *wejście* to wartość Cb , Cr .

org



$YCbCr$, kolory * 1.5, RGB



porównanie v.PS * 10



porównanie v.2 * 10

