

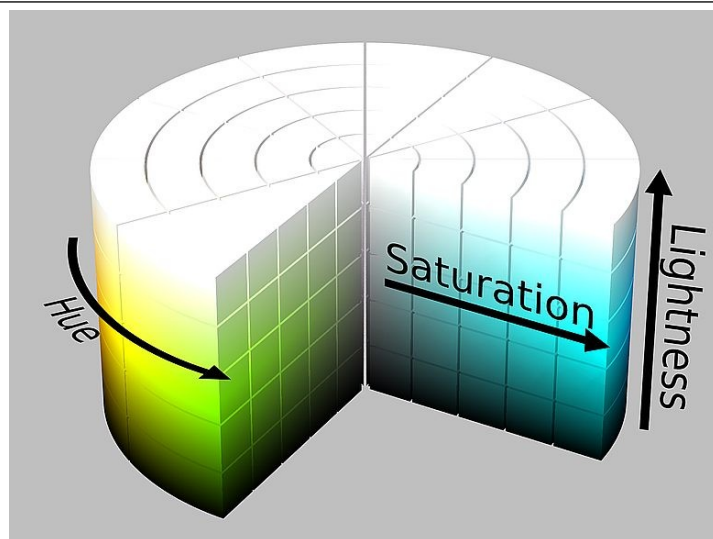
Proszę wykonać oddzielnie podane niżej zadania dla obrazka „rzeczka.jpg” lub „rzeczka_mniejsza.jpg”. Nie należy korzystać z gotowych funkcji dokonujących obróbki obrazka. Podane niżej przykłady zostały zrobione dla obrazka „rzeczka.jpg”, ale będą one bardzo podobne dla obróbki „rzeczka_mniejsza.jpg”, a obróbka mniejszej wersji powinna być znacznie szybsza.



Teoria dot. HSL

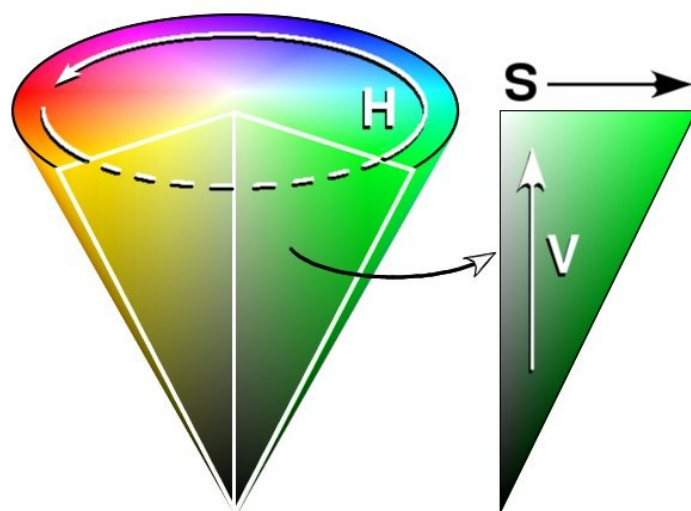
HSL to specjalna przestrzeń trójwymiarowa przeznaczona do prezentacji koloru danego piksela. Piksel opisywany jest przez 3 składowe:

- **Hue** (? barwa ?) – jest to miara opisująca jaki kolor mamy do czynienia, jej zakres to [0; 360] stopni. Miara ta ma charakter kołowy, dlatego piksele o Hue 359 i 1 mają bardzo zbliżony kolor (różnica tylko 2 stopni). Hue dla jasności (Lightness) maksymalnej i minimalnej jest bez znaczenia, ponieważ piksel jest idealnie biały albo czarny. Hue dla nasycenia (Saturation) równego zero jest również nie istotna, ponieważ oznacza piksel w odcieniach idealnej szarości. Warto zwrócić uwagę, że w wyniku zastosowanych filtrów Hue może przekroczyć zakres [0; 360], w takim przypadku należy dodać lub odjąć wielokrotność 360 stopni tak, aby po przetworzeniu Hue mieściło się w [0; 360] stopni, na przykład 730 -> 10; -20 -> 340.
- **Saturation** (nasycenie kolorów) – miara ta opisuje nasycenie kolorów, jej zakres to [0; 1]. Nasycenie dla jasności (Lightness) maksymalnej i minimalnej jest bez znaczenia, ponieważ piksel jest idealnie biały albo czarny, pomimo tego dla tych przypadków nasycenie powinno wynosić 0.
- **Lightness** (jasność) – miara ta określa jasność i jej zakres to [0; 1].



Przestrzeń HSL, źródło: wikipedia

Istnieje również podobna przestrzeń barw do HSL nazwana HSV:



Przestrzeń HSV, źródło wikipedia

Również sposób przetwarzania z RGB na HSV i odwrotnie jest bardzo podobny do HSL.

1. Proszę przetworzyć obrazek z przestrzeni RGB na HSL zgodnie z podanym niżej kodem, który dotyczy każdego pojedynczego piksela osobno. Po dokonanych przetworzeniach proszę wyświetlić poszczególne składowe.

- Niech składowe R, G, B opisują poszczególne składowe koloru pojedynczego piksela.
- Przeskaluj wartości R, G, B do zakresu [0; 1].
- Wylicz wartości $MaxRGB = \max\{R, G, B\}$, $MinRGB = \min\{R, G, B\}$.
- Oblicz $Chroma = MaxRGB - MinRGB$.

- Wylicz H (Hue, barwa) zgodnie ze wzorem

$$H_{temp} = \begin{cases} 0 & \text{jeśli } Chroma \leq 0,001, \\ \text{modulo}\left(\frac{G-B}{Chroma}, 6\right) & \text{jeśli } R = \text{MaxRGB} \wedge Chroma > 0,001 \\ \left(\frac{B-R}{Chroma}\right) + 2 & \text{jeśli } G = \text{MaxRGB} \wedge Chroma > 0,001 \\ \left(\frac{R-G}{Chroma}\right) + 4 & \text{jeśli } B = \text{MaxRGB} \wedge Chroma > 0,001 \end{cases}$$

$$H = H_{temp} * 60$$

- Ustal L (jasność) zgodnie ze wzorem $L = \frac{\text{MaxRGB} + \text{MinRGB}}{2}$

- Oblicz S (nasycenie kolorów) $S = \begin{cases} 0 & \text{jeśli } L \leq 0,001 \vee L \geq 0,999, \\ \frac{Chroma}{1 - |2 * L - 1|} & \text{w pozostałym przypadku} \end{cases}$

RGB



Hue / 360stopni



Saturation



Lightness



Warto napisać kod przetwarzania bez użycia pętli w celu optymalizacji.

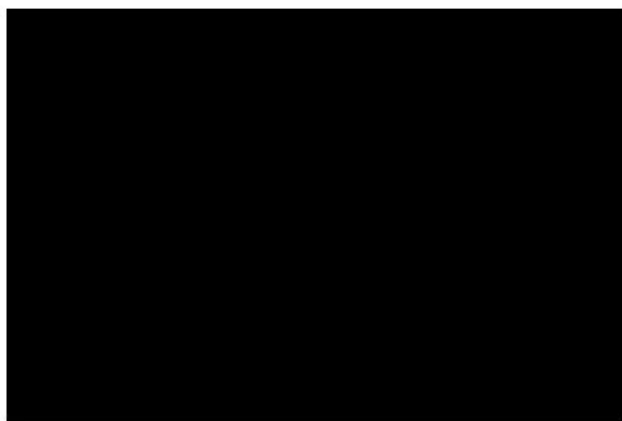
2. Proszę przetworzyć obrazek z przestrzeni RGB, na HSL, później znowu na RGB i porównać jakie zostały otrzymane błędy/niezgodności. Proces konwersji z HSL do RGB jest widoczny na poniższym kodzie, dotyczy on każdego piksela osobno.

- Trzeba wyliczyć $Chroma = (1 - |2 * L - 1|) * S$.
 - Następnie zmienną tymczasową $X = Chroma * \left(1 - \left| \text{modulo} \left(\frac{H}{60}, 2 \right) - 1 \right| \right)$.
 - Później wyliczyć $MinRGB = L - \frac{Chroma}{2}$.
 - Kolejno wyliczyć R, G, B
- $$R' = \begin{cases} Chroma & \text{jeśli } 0 \leq H \leq 60 \\ X & \text{jeśli } 60 < H \leq 120 \\ 0 & \text{jeśli } 120 < H \leq 180 \\ 0 & \text{jeśli } 180 < H \leq 240 \\ X & \text{jeśli } 240 < H \leq 300 \\ Chroma & \text{jeśli } 300 < H \leq 360 \end{cases}, \quad G' = \begin{cases} X & \text{jeśli } 0 \leq H \leq 60 \\ Chroma & \text{jeśli } 60 < H \leq 120 \\ Chroma & \text{jeśli } 120 < H \leq 180 \\ X & \text{jeśli } 180 < H \leq 240 \\ 0 & \text{jeśli } 240 < H \leq 300 \\ 0 & \text{jeśli } 300 < H \leq 360 \end{cases},$$
- $$B' = \begin{cases} 0 & \text{jeśli } 0 \leq H \leq 60 \\ 0 & \text{jeśli } 60 < H \leq 120 \\ X & \text{jeśli } 120 < H \leq 180 \\ Chroma & \text{jeśli } 180 < H \leq 240 \\ Chroma & \text{jeśli } 240 < H \leq 300 \\ X & \text{jeśli } 300 < H \leq 360 \end{cases},$$

$$R = R' + MinRGB; G = G' + MinRGB; B = B' + MinRGB .$$

- Przeskalować R, G, B z zakresu [0; 1] na [0; 255]
- Oto wynik działania porównania z obrazka przetworzonego w obie strony z oryginałem:

porównanie v.PS * 1000000



porównanie v.2 * 10000



- maksymalna różnica między oryginałem a obrazkiem przetworzonym na HSL i z powrotem powinien być około $5e-14$

3. Proszę przetworzyć obrazek z RGB na HSL, następnie odjąć od składnika Hue (barwa) 50 i przetworzyć z powrotem na RGB. Ponieważ

składnik Hue określa kąt i powinien się on mieścić w przedziale $[0; 360)$, dlatego wartości mniejsze niż 0 powinny być powiększone o 360.

org



HUE - 50



4. Proszę przetworzyć obrazek z RGB na HSL, następnie powiększyć wartość nasycenia kolorów (składnik Saturation) filtrem gamma 1.5 i przetworzyć z powrotem na RGB.

org



Saturation gamma 1.5



5. Proszę przetworzyć obrazek z RGB na HSL, następnie zmniejszyć jasność (składnik Lightness) filtrem gamma 0,7 i przetworzyć z powrotem na RGB.

org



Lightness gamma 0.7

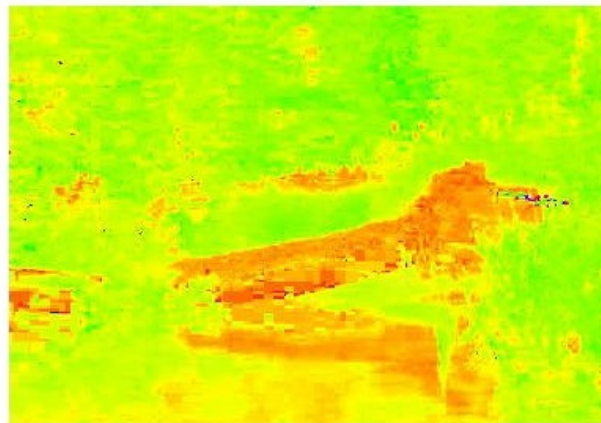


6. Proszę przetworzyć obrazek z RGB na HSL, następnie dla wszystkich pikseli o nasyceniu kolorów > 0 ustawić jasność (składnik Lightness) na 0,5 i nasycenie (składnik Saturation) na 1, po tym przetworzyć z powrotem na RGB.

org



Jesli Sat.>0, Lightness=0.5 i Saturation=1



7. Dla ambitnych. Proszę przetworzyć obrazek z RGB na HSL, następnie w płynny sposób zmniejszyć nasycenie kolorów i jasność (przez zwykłe pomnożenie przez wartości płynnie zmieniające się w przedziale $[0; 1]$) tak, aby uzyskać efekt widoczny poniżej, po tym należy przetworzyć z powrotem na RGB.

S,L gradienty * (0->1)

