Лекція 5

Представлення кольору. RGB та HSV

Навіщо робити перетворення кольору?

Ця тема потрібна для ясного засвоєння гістограм та фільтрації.

Важливість її обумовлена тим, що при гістограмних перетвореннях колір кожного пікселя змінюється незалежно від його сусідів, а при фільтрації – в залежності від кольорів сусідніх пікселів.

Математично такі залежності виконуються за допомогою векторних перетворень. Якщо потрібно щось обчислювати з векторами, то сучасна *обчислювальна* математика оперує матрицями – можна сказати, що замість векторів. Насправді вектори у просторі можуть бути описані матрицями, і тоді ***всі*** операції з векторами зводяться до операцій з матрицями.

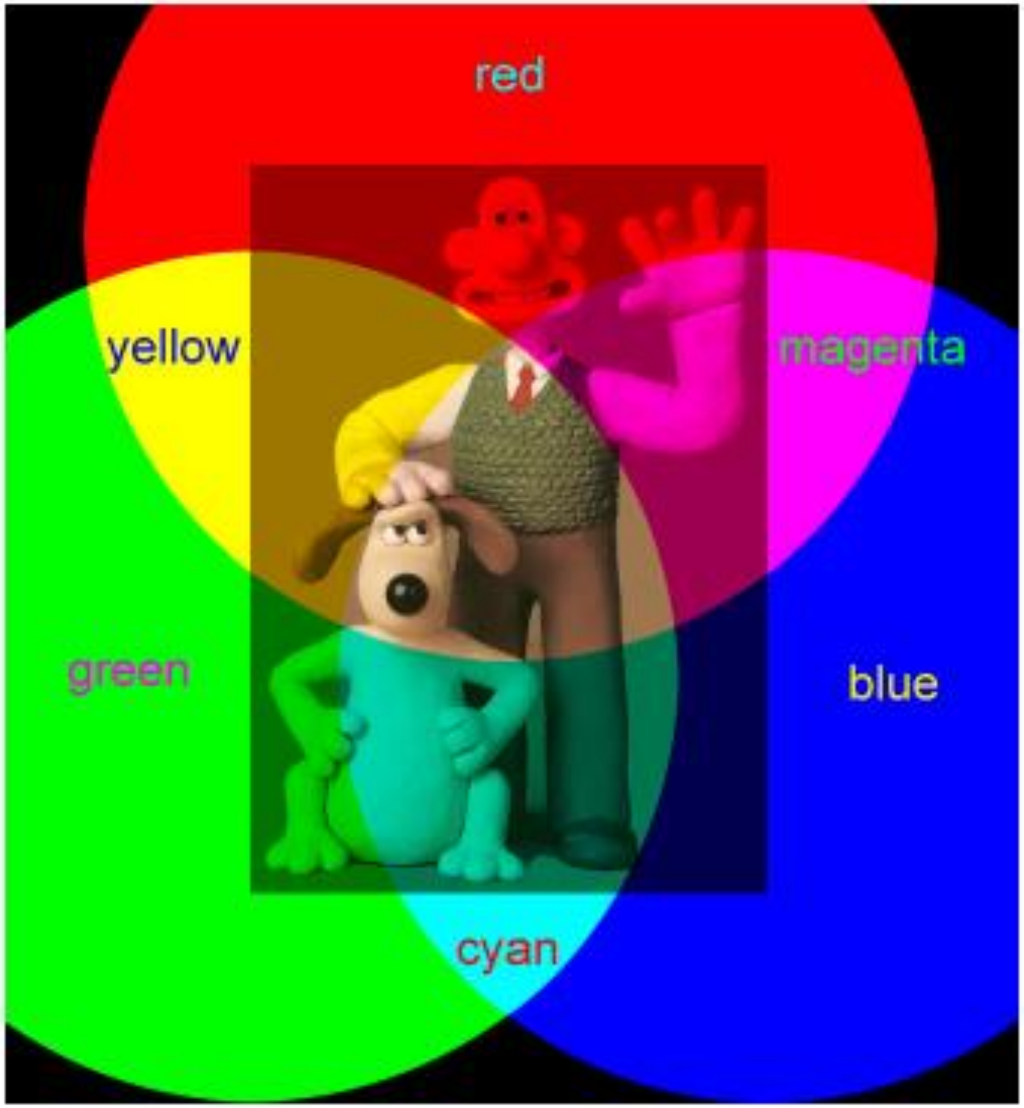
Основна мета цієї теми – показати, чому колір на зображенні – це вектор; якими способами можна описати цей вектор математично і як це буде виглядати “на кольорі”; що можна робити з цим вектором і як це буде обчислюватися.

Колірний простір RGB

Кожен піксель зображення кодується трьома байтами: перший відображає насиченість (інтенсивність) червоної складової, другий – зеленої, третій – синьої. Практично по цій самій моделі працює людське око (клітини, які відповідають за кольорове бачення (так звані *колбочки*) є трьох видів, і вони реагують саме на ці колірні складові світла.

Змішуючи червоний із зеленим, отримуємо жовтий колір, зелений з синім – блакитний, а синій з червоним – фіолетовий.

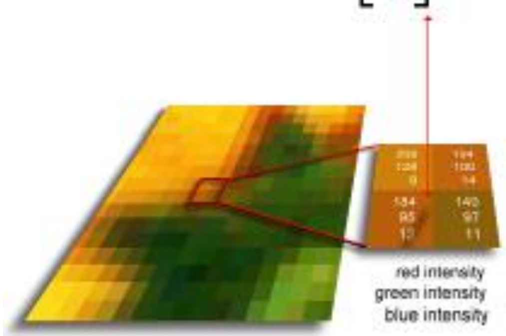
Червоний, зелений і синій називаються ***основними*** кольорами, а блакитний, фіолетовий та жовтий – ***додатковими***.

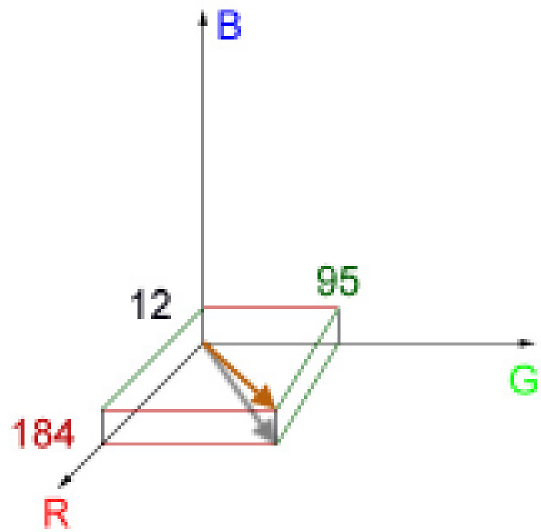


Колірний простір RGB
Колір – це вектор

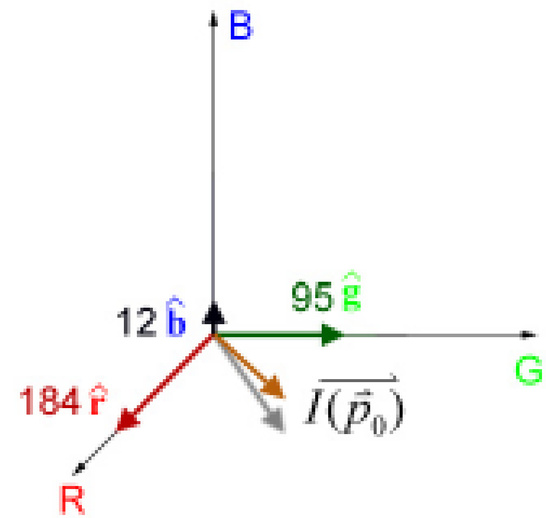
Якщо взяти три складові кольору, то загальний колір можна представити як вектор у тривимірному просторі. Такий вектор записується як матриця-стовпчик, а ще його можна розписати по базису.

$$I(\vec{p}_0) = \begin{bmatrix} 184 \\ 95 \\ 12 \end{bmatrix}$$





$$I(\vec{p}_0) = \begin{bmatrix} r_0 \\ g_0 \\ b_0 \end{bmatrix} = \begin{bmatrix} 184 \\ 95 \\ 12 \end{bmatrix}$$



$$\overrightarrow{I(\vec{p}_0)} = r_0\hat{\mathbf{r}} + g_0\hat{\mathbf{g}} + b_0\hat{\mathbf{b}}$$
$$= 184\hat{\mathbf{r}} + 95\hat{\mathbf{g}} + 12\hat{\mathbf{b}}$$

Колірний простір RGB
Властивості

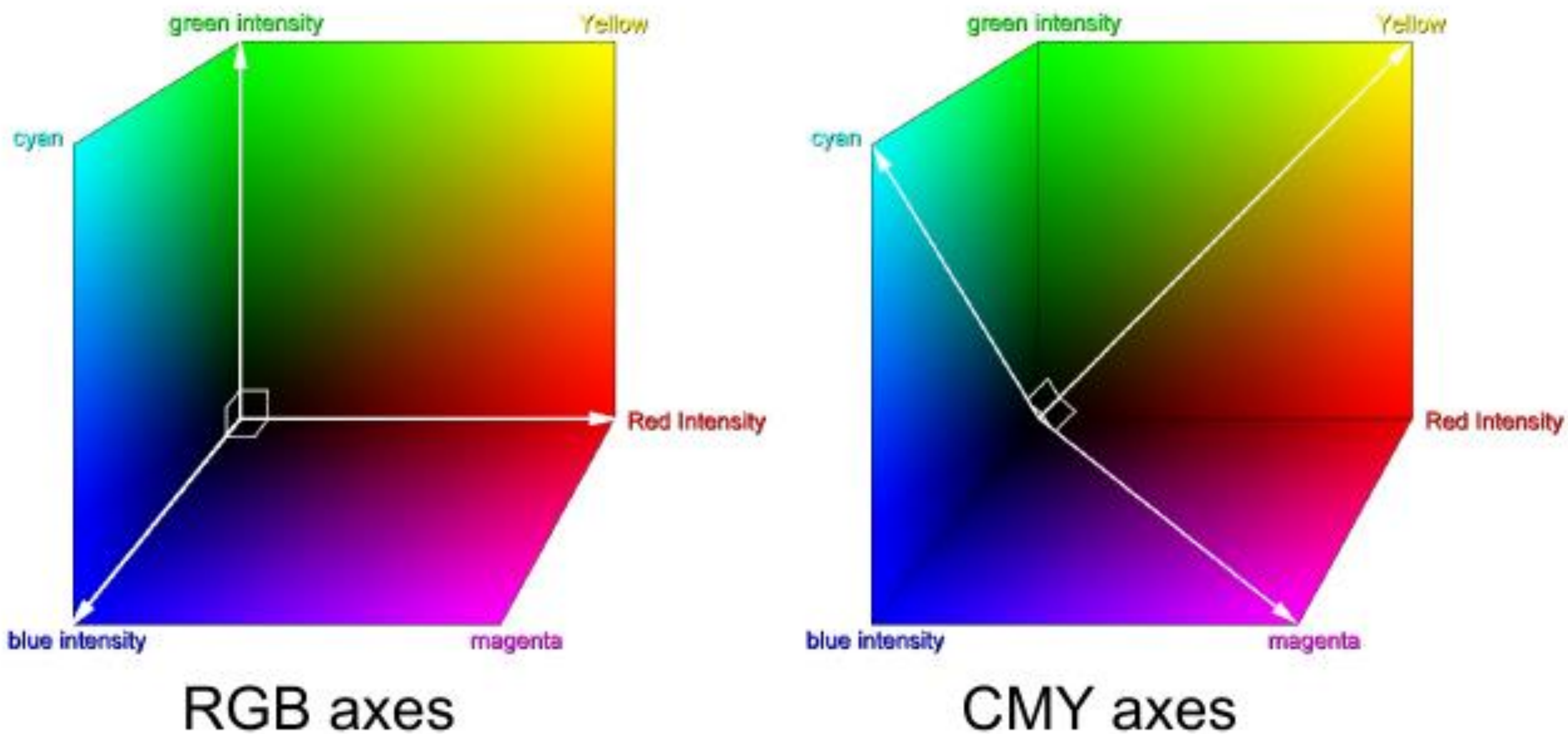
1. Кольори (R, G, B) відповідають осям цього простору.

2. Інтенсивності кольору по кожній вісі мають 256 градацій, причому значення 0 відповідає відсутності кольору.

3. Немає від’ємних значень – тобто колірний простір відповідає першому октанту повного трьохвимірного простору.

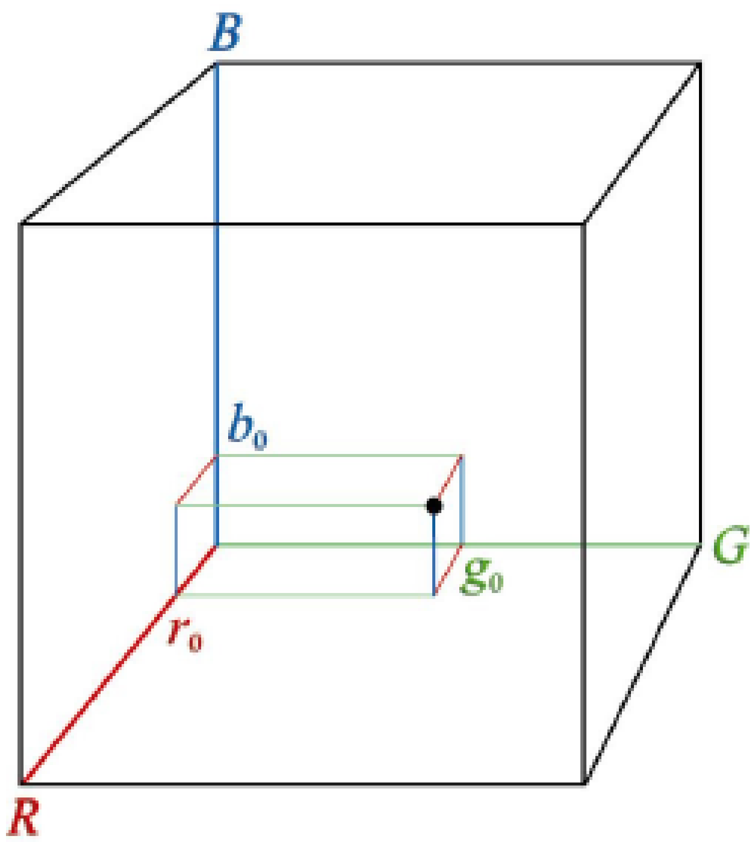
4. Колірний простір дискретний, якщо по кожній вісі вкладаються значення від 0 до 255, то загальна кількість кольорів становить 256^3 = 16 777 216.

Колірний простір RGB можна якби “повернути на 45°”, і таким чином отримати колірний простір CMY, у якому вздовж осей будуть інтенсивності додаткових кольорів – блакитного (Cyan), фіолетового (Magenta) та жовтого (Yellow).

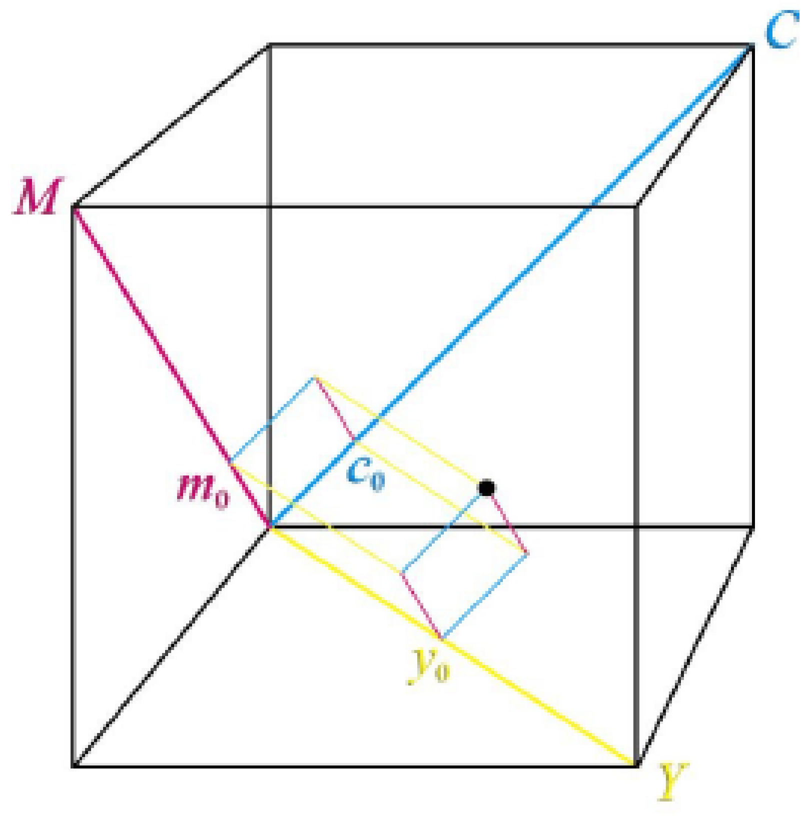


Як це виглядає на векторах…

RGB:

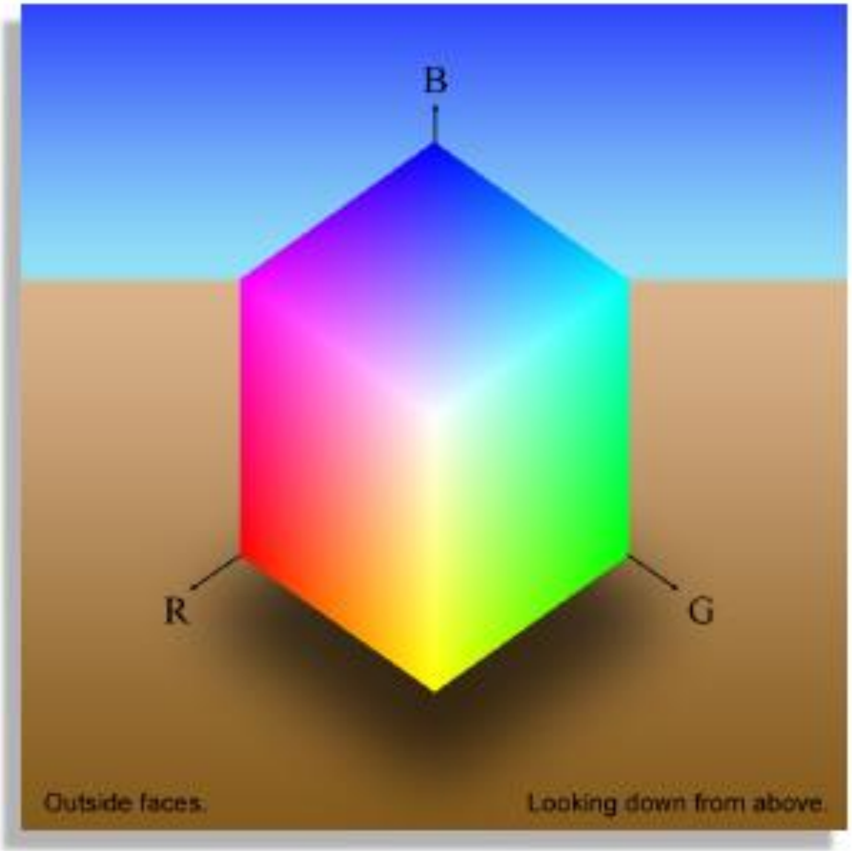


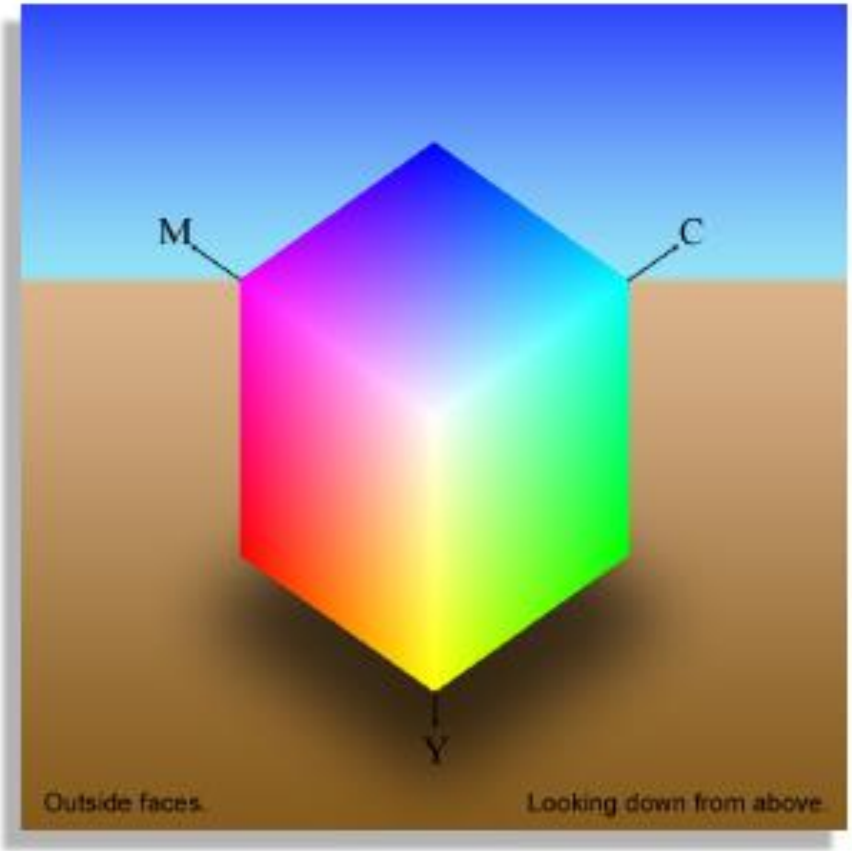
CMY:



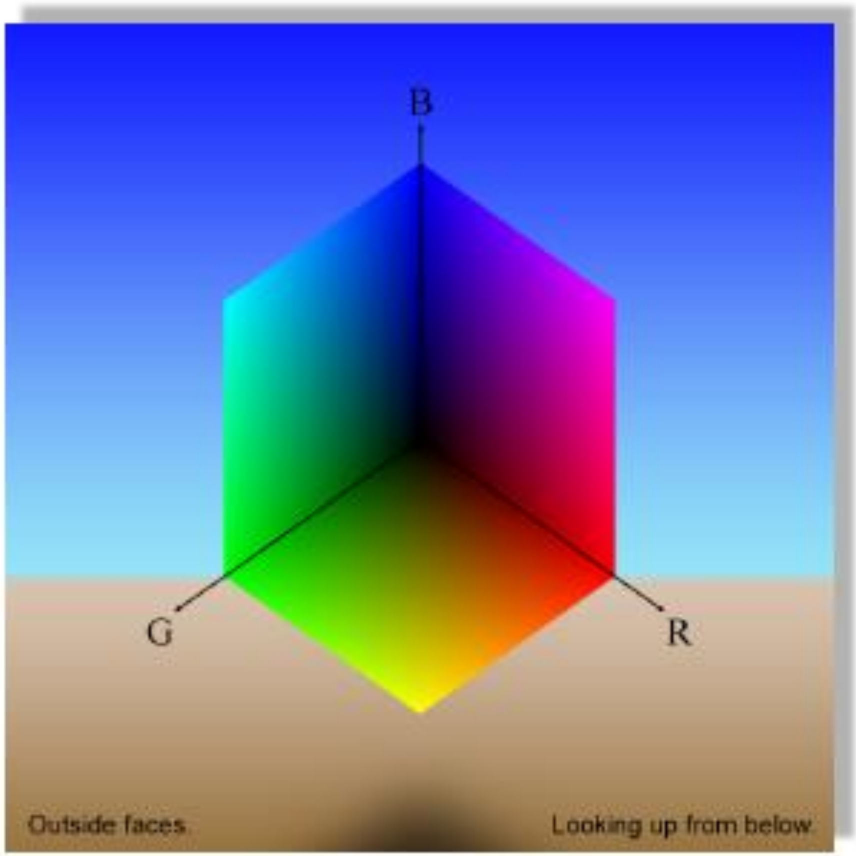
Колірні простори RGB та CMY
…і як це виглядає на кольорі

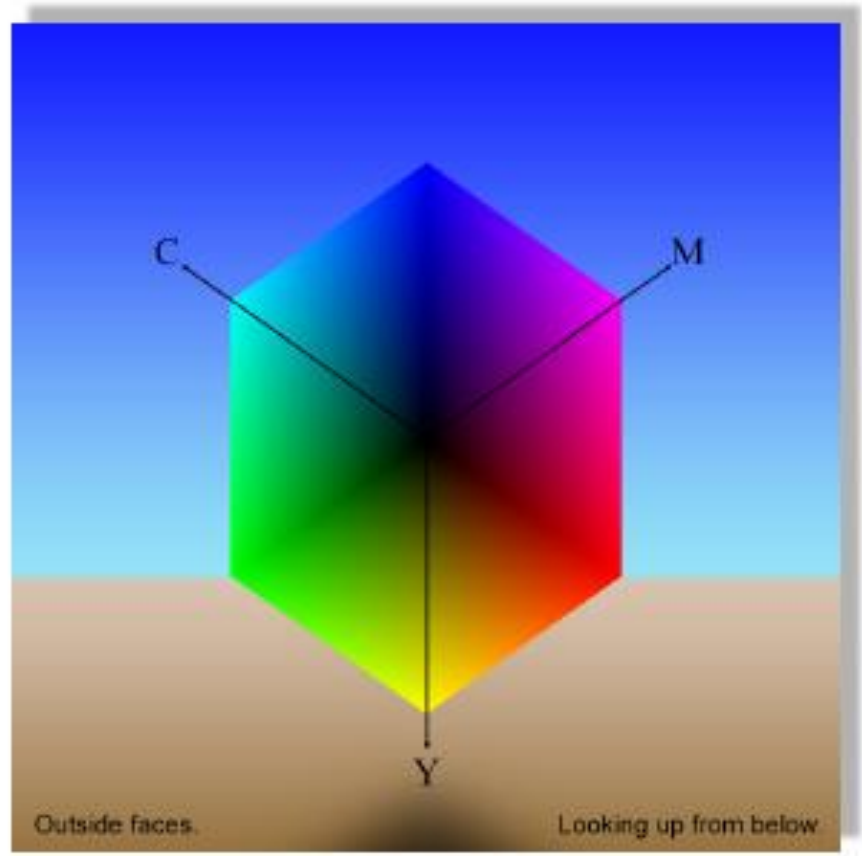
В колірному просторі RGB точка (0; 0; 0) відповідає чорному кольору, а точка (255; 255; 255) – білому. Так виглядає “кольоровий RGB-куб” з боку білої точки.
(Рисунки, подібні до цих можна розглядати як 3D – для цього на нього треба дивитися прямо і розфокусувати погляд)



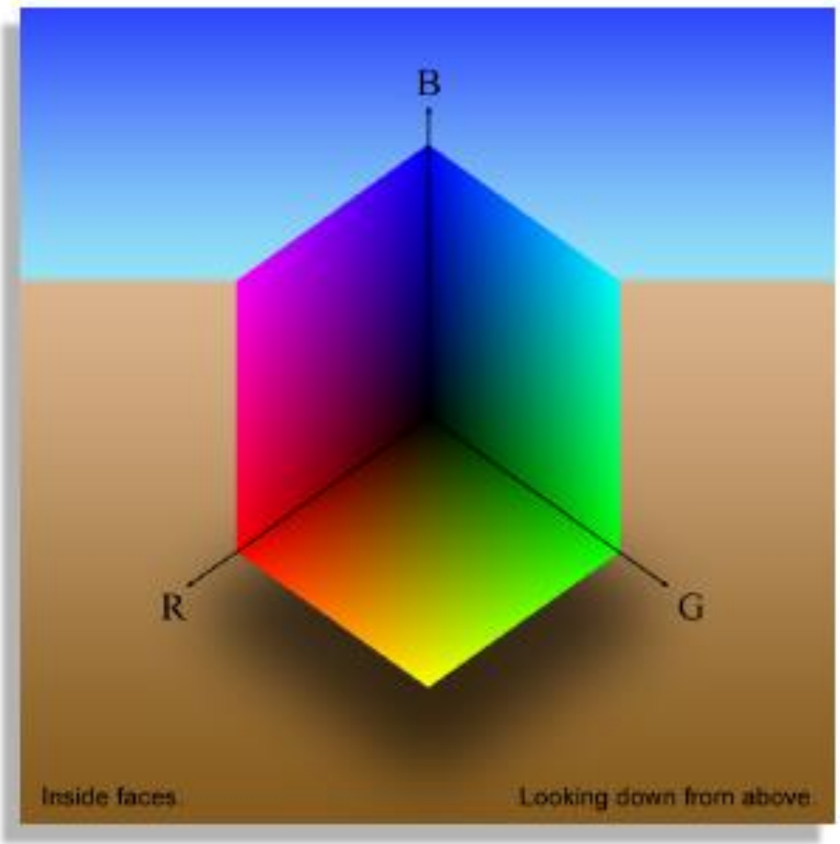


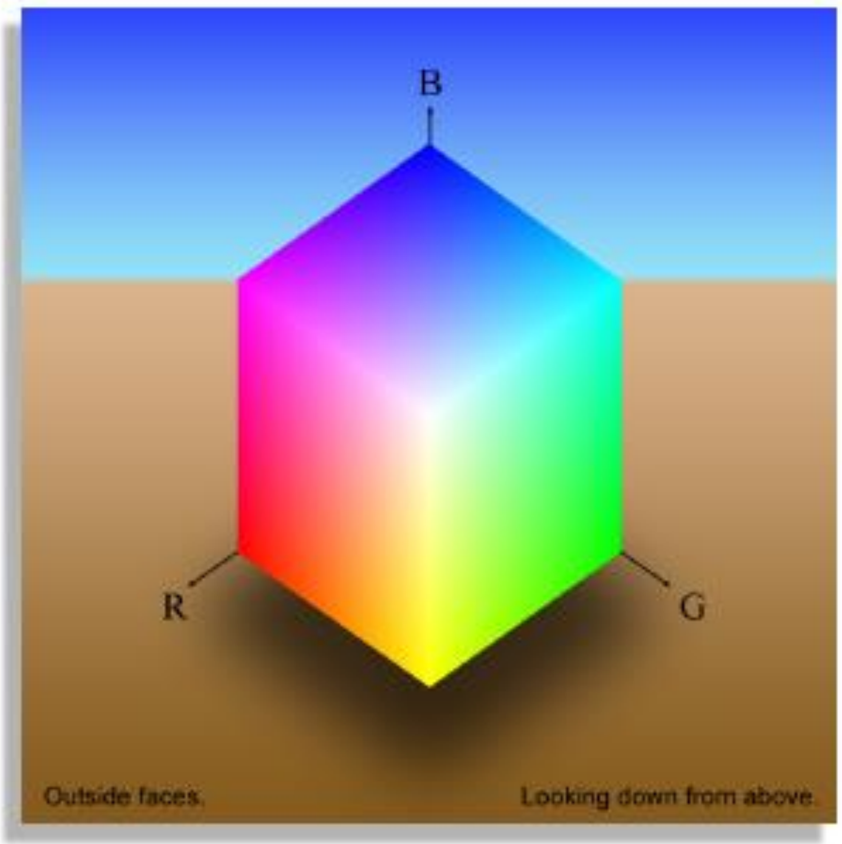
Колірний простір RGB
З боку чорної точки





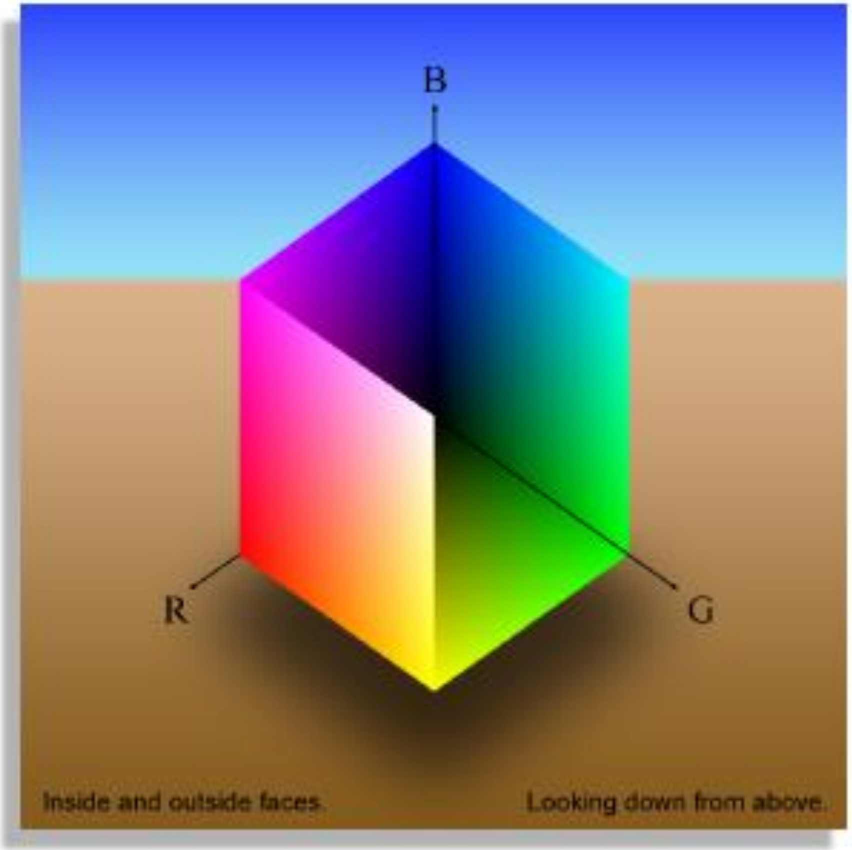
Колірний простір RGB
Зсередини та зовні*

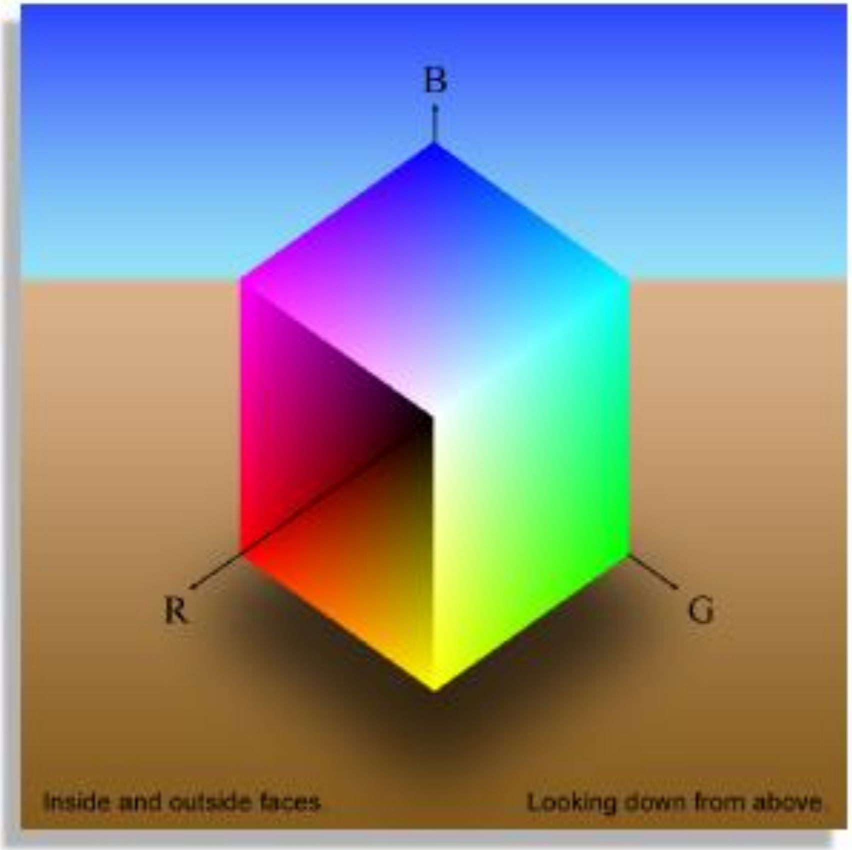




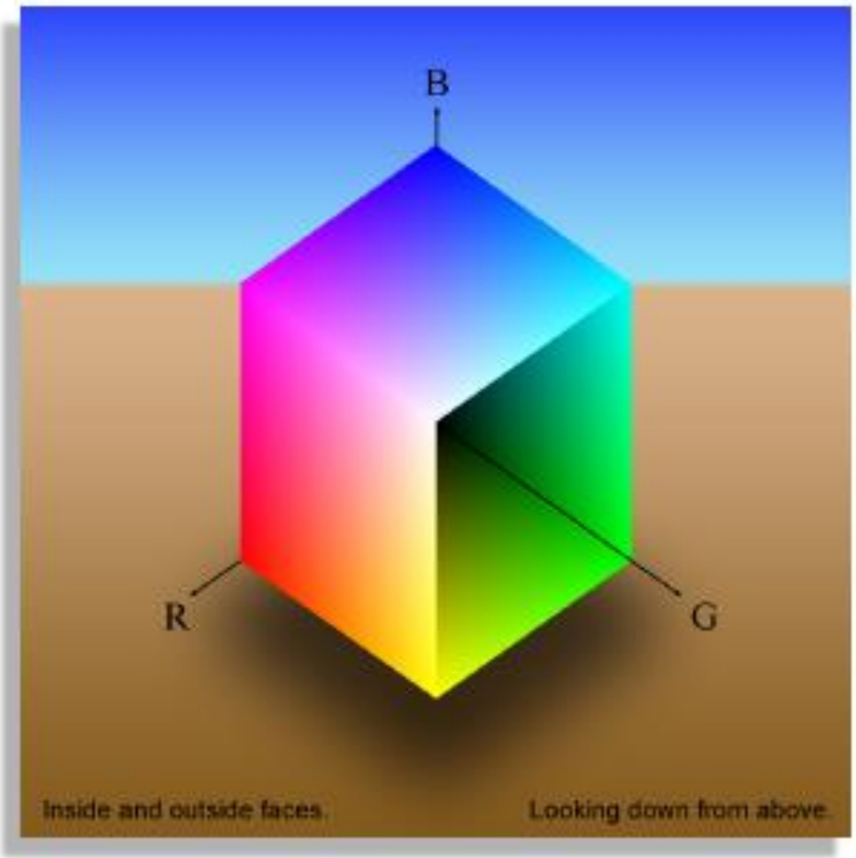
*) Обережно, це не зовсім 3D-картинка… Намагатися розглядати як 3D-картинку можна, але на свій страх і ризик… проте, якщо получиться, можна поміркувати над тим, який куб ви бачите – зовні чи зсередини… і з боку якої точки – білої чи чорної.

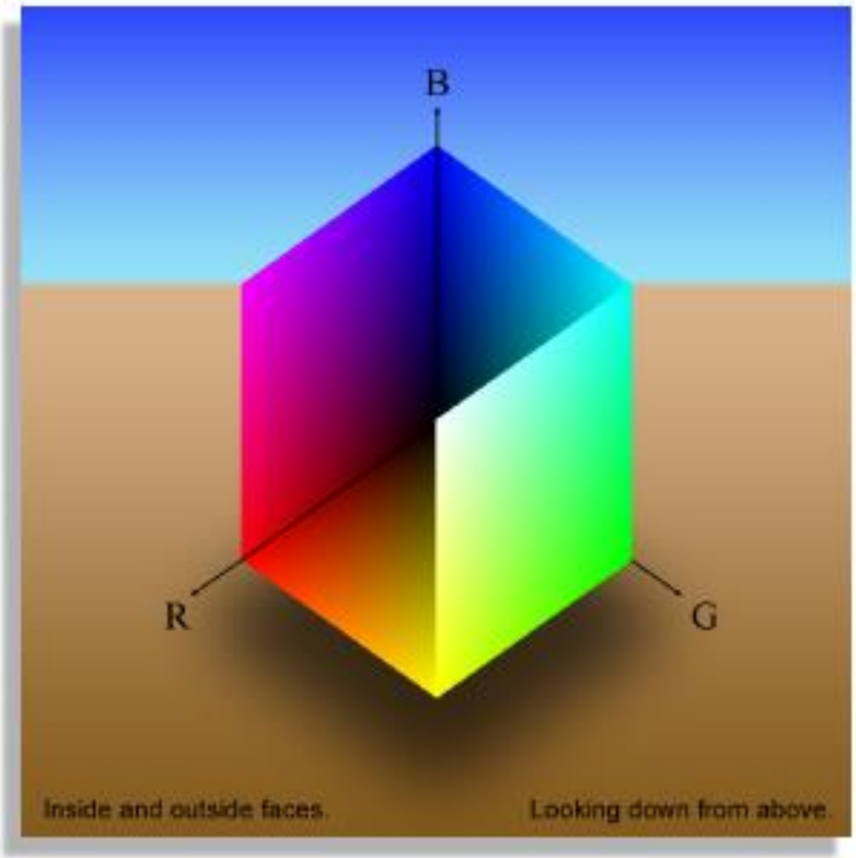
Колірний простір RGB
Зсередини та зовні



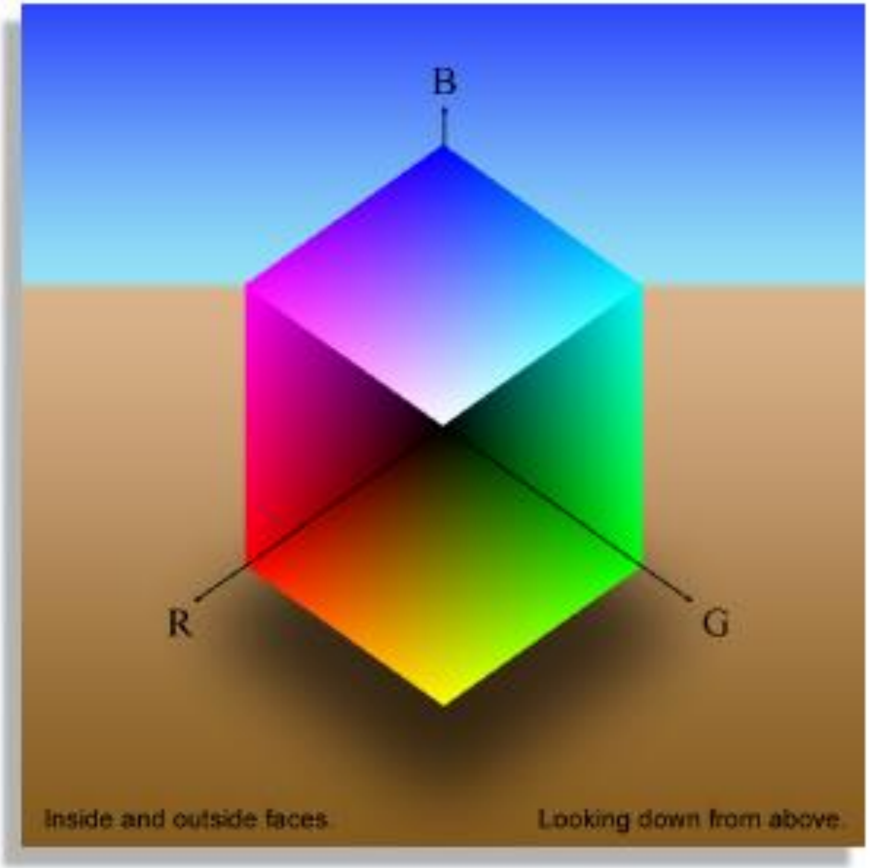


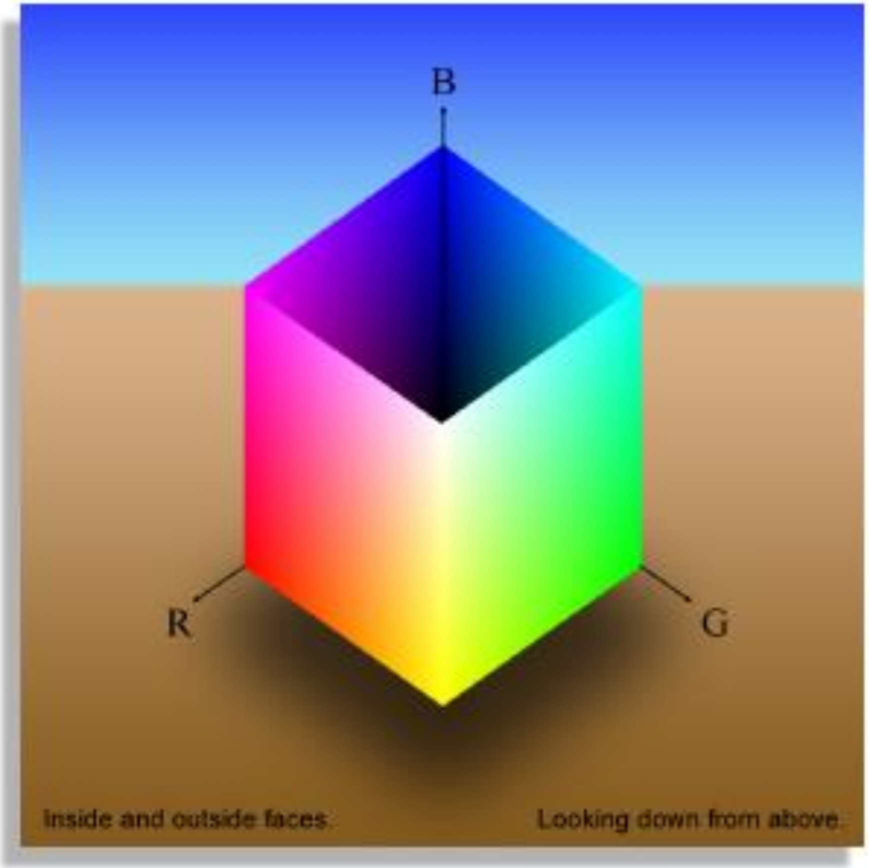
Колірний простір RGB
Зсередини та зовні





Колірний простір RGB
Зсередини та зовні





Колірний простір HSV

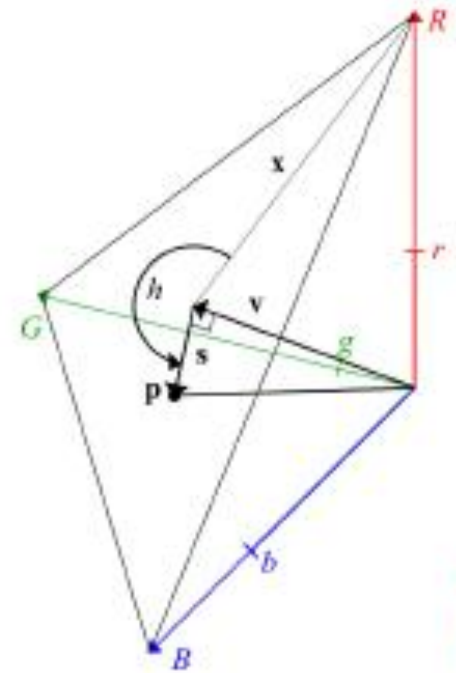
RGB – це найвідоміший спосіб представлення кольорів, саме цей спосіб використовується у цифрових пристроях візуалізації для кодування та відображення зображень.

Проте для аналізу вектори колірного простору RGB можуть бути перетворені у вектори інших колірних просторів. Тут ми розглянемо одне таке представлення - кольоровий простір HSV. Це представлення кольору в циліндричних координатах, які відокремлюють яскравість (значення, *Value*) від кольоровості (відтінок (*Hue*) і насиченість (*Saturation*)).

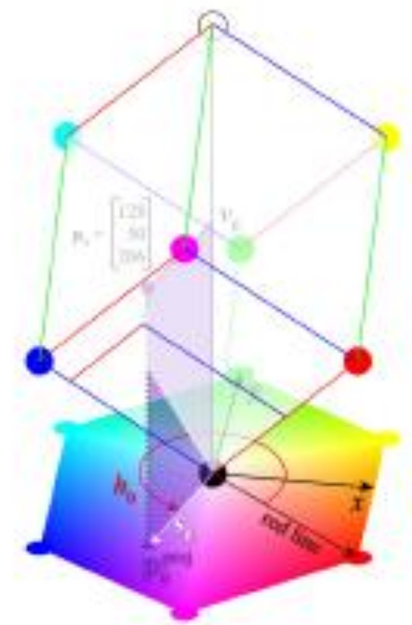
Таке представлення особливо актуальне при фільтрації та сегментації зображень, тому що замість обчислень на основі трьох значень кольору (RGB) можна робити обчислення по якомусь одному (при сегментації по яскравості (Value), при фільтрації – по відтінку або насиченості).

Є два типи алгоритмів перетворення RGB в HSV.

Перший - векторно-геометричний алгоритм – є математично точним, але непрактичним для швидких обчислень.



Другий тип алгоритмів базується на основі представлення кольору RGB як проекції на кольоровий шестикутник. Він працює приблизно в 3 рази швидше, ніж перший, і саме він найчастіше використовується на практиці.



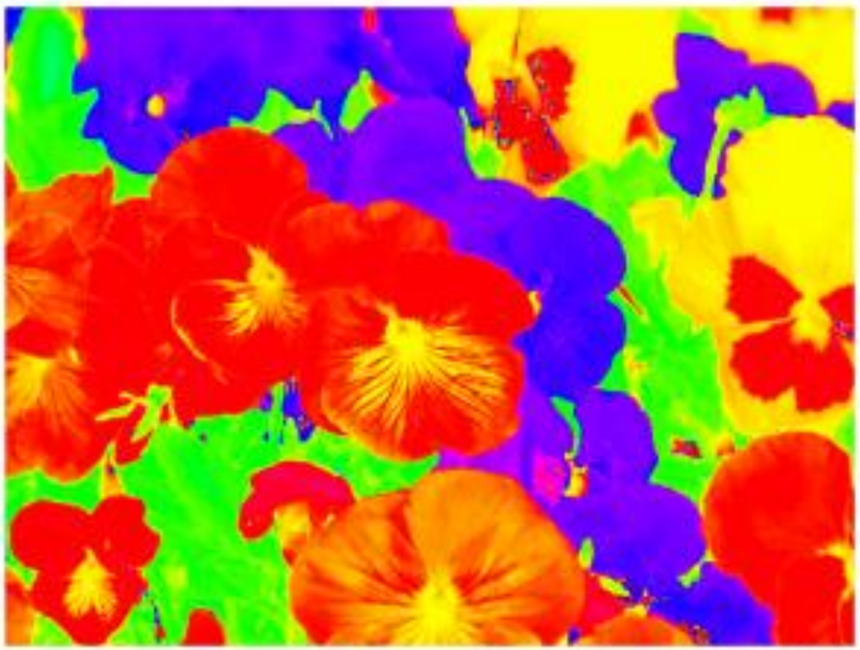
Обидва алгоритми дають однакові відтінки, але їх компоненти насичення та значення відрізняються.

Яскравість кольору (*Value*)

Зображення, кодоване якравістю (значенням, Value) - це монохромне зображення, колір кожного пікселя обчислюється як середнє значення трьох кольорів у кожному пікселі.

Відтінок та насиченість (*Hue and saturation*)

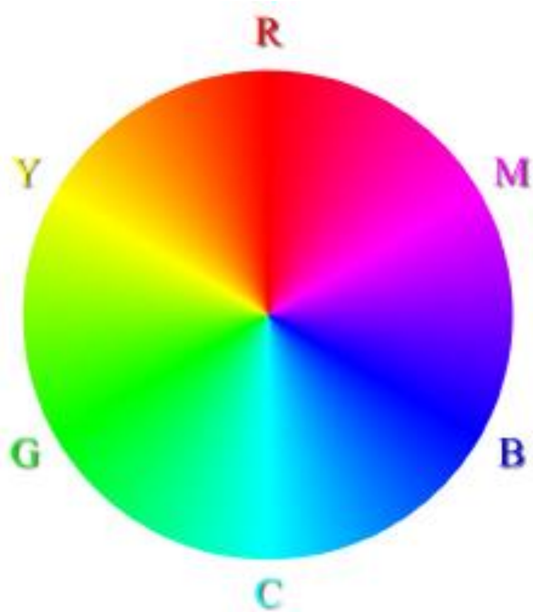
Інформація про колір в зображенні, яку іноді називають поширюваною, має два виміри: відтінок і насиченість.

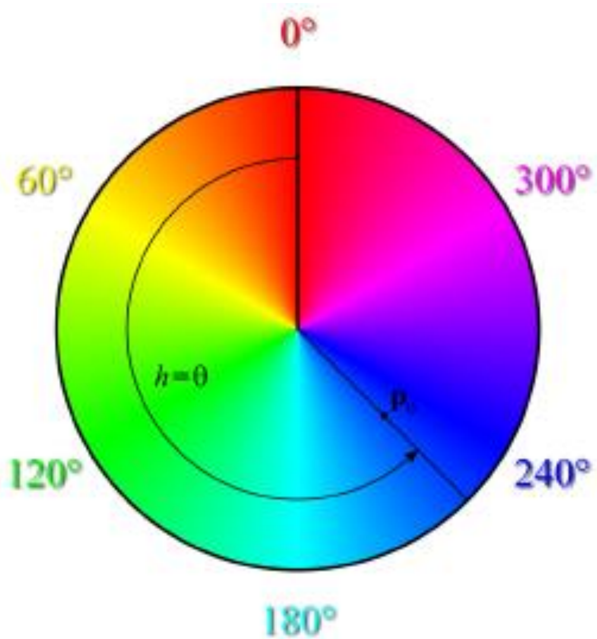


Відтінок (*Hue*)

Нагадаємо, що клітини кольорових рецепторів сітківки (*колбочки*) чутливі до трьох різних довжин хвиль світла. Сукупні відповіді цих клітин дають нам сприйняття відтінку, який характеризується так званим *кольоровим колом (color wheel)*.

Будь-який відтінок кола може бути представлений чисельно, як виміряний кут проти годинникової стрілки від опорного відтінку. Традиційно червоний колір використовується як опорний і йому присвоюється кут 0°. Кожен інший відтінок має свій кут, $h = \theta \in [0, 360)$.

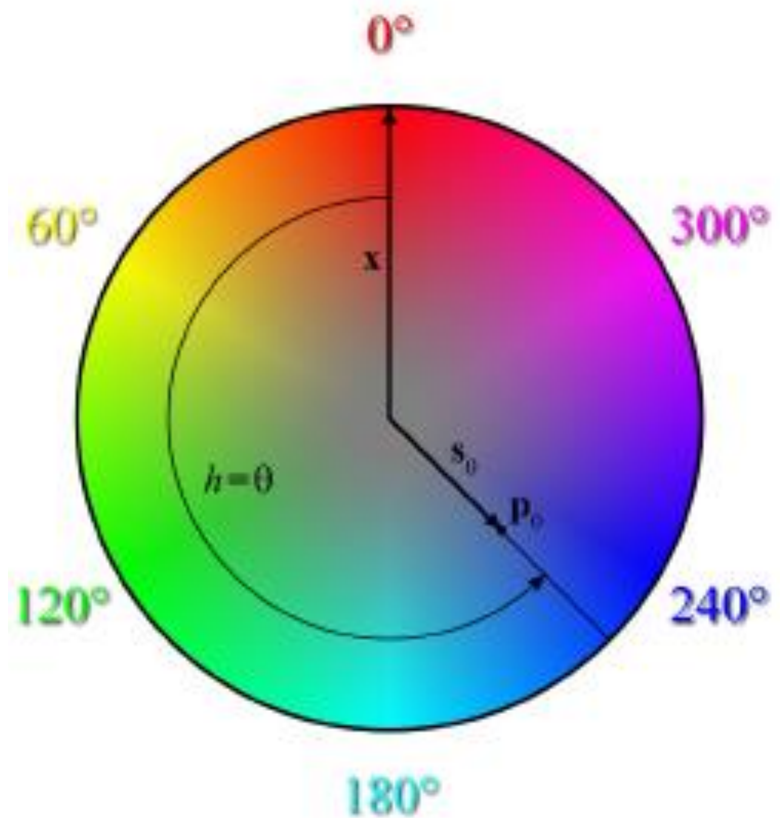




Насиченість (*Saturation*)

Насиченість представляється у кольоровому колі як відстань від його центру. У центрі насиченість 0. У насиченості немає кольору, просто інтенсивність сірого значення. Насиченість радіально зростає до максимуму - одиниці. Одиниця відповідає чистому відтінку.

Простіше кажучи, колірний простір HSV – це циліндрична система координат, в якій традиційні координата *z* – це яскравість (або значення, *Value*), координата *φ* (кут) – відтінок (*Hue*), а координата *ρ* (радіус) – насиченість *(Saturation)*.



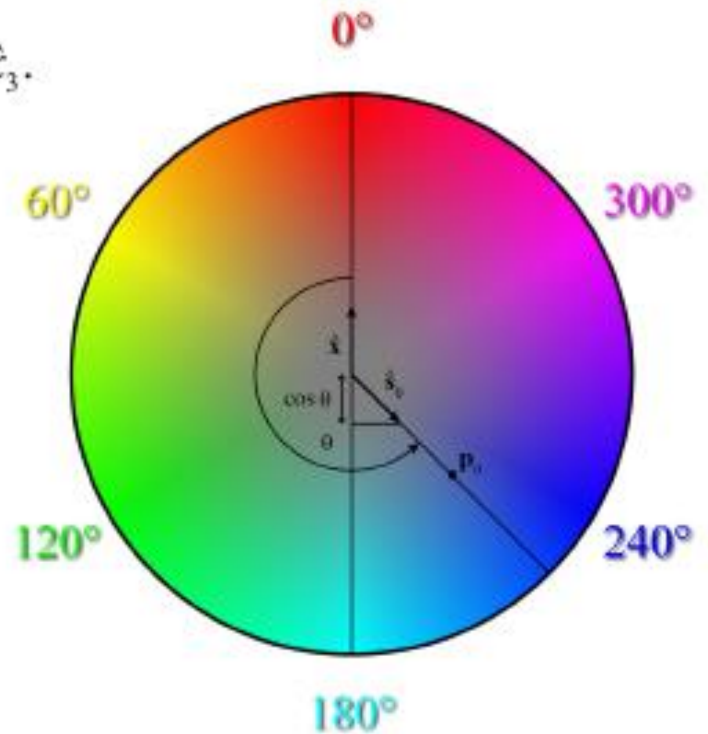
Колірний простір HSV
Обчислення

Ми можемо обчислити кут відтінку через косинус кута, заданого двома одиничними векторами із центра кола: x у напрямку червоного та s_0 у напрямку кольору p_0. Ці одиничні вектори обчислюються діленням векторів x і s_0 на їх відповідну довжину.

$$\cos\theta = \hat{\mathbf{s}} \bullet \hat{\mathbf{x}} = \hat{s}_1\hat{x}_1 + \hat{s}_2\hat{x}_2 + \hat{s}_3\hat{x}_3.$$

$$\hat{\mathbf{s}}_0 = \begin{bmatrix} \hat{s}_1 \\ \hat{s}_2 \\ \hat{s}_3 \end{bmatrix}, \quad \hat{\mathbf{x}} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{bmatrix}$$

All the vectors have 3 components. The first 2 are the rectangular coordinates of the vector in the plane of the wheel the third is the value of color p_0.



$$\hat{\mathbf{s}}_0 = \frac{\mathbf{s}_0}{\|\mathbf{s}_0\|}, \quad \hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$$

$$\mathbf{s}_0 = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\|\mathbf{s}_0\| = \sqrt{s_1^2 + s_2^2 + s_3^2}$$

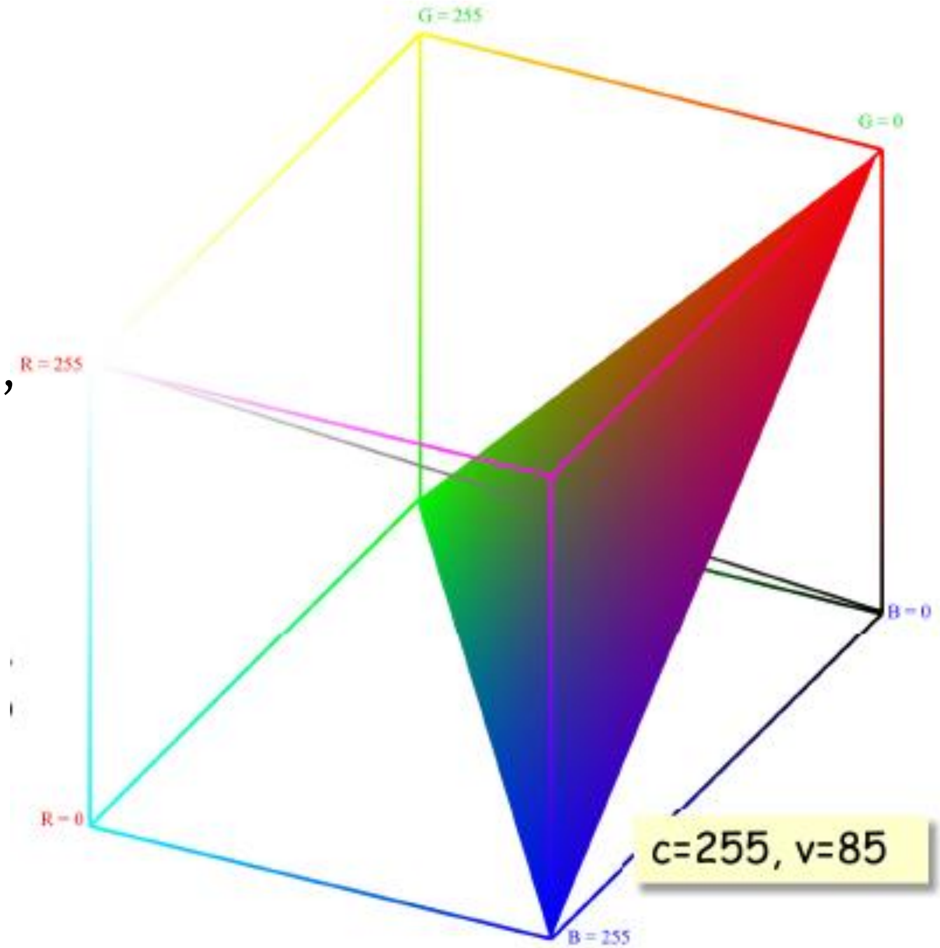
$$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + x_3^2}$$

Еквівалентний кольоровий трикутник

Три вектори у просторі з координатами

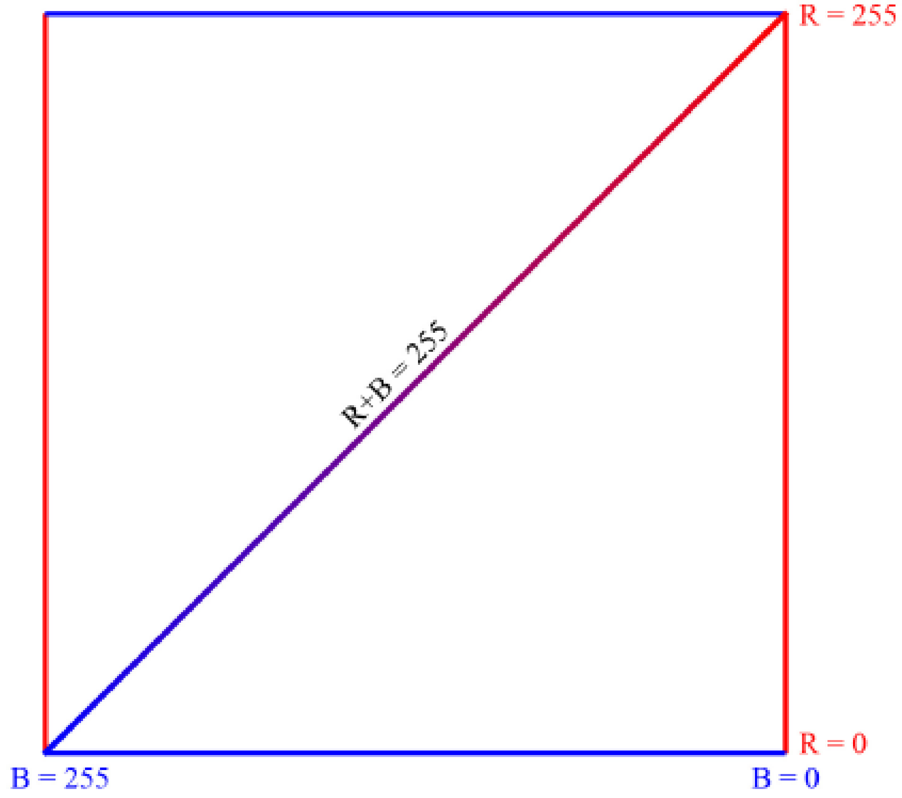
$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ c \\ 0 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix},$$

утворюють трикутник всередині кольорового куба, якщо $c \leq 255$ або $c \geq 510$, або шестикутник, якщо $255 < c < 510$. Кожен колір на площинній поверхні має $r + g + b = c$. Тому його яскравість є $c/3$. Площина такого трикутника (або шестикутника) перпендикулярна до сірої лінії ($c[1,1,1]$). Якщо дано значення кольору, то значення відтінку та насиченості обчислюються в відповідній еквівалентній площині.



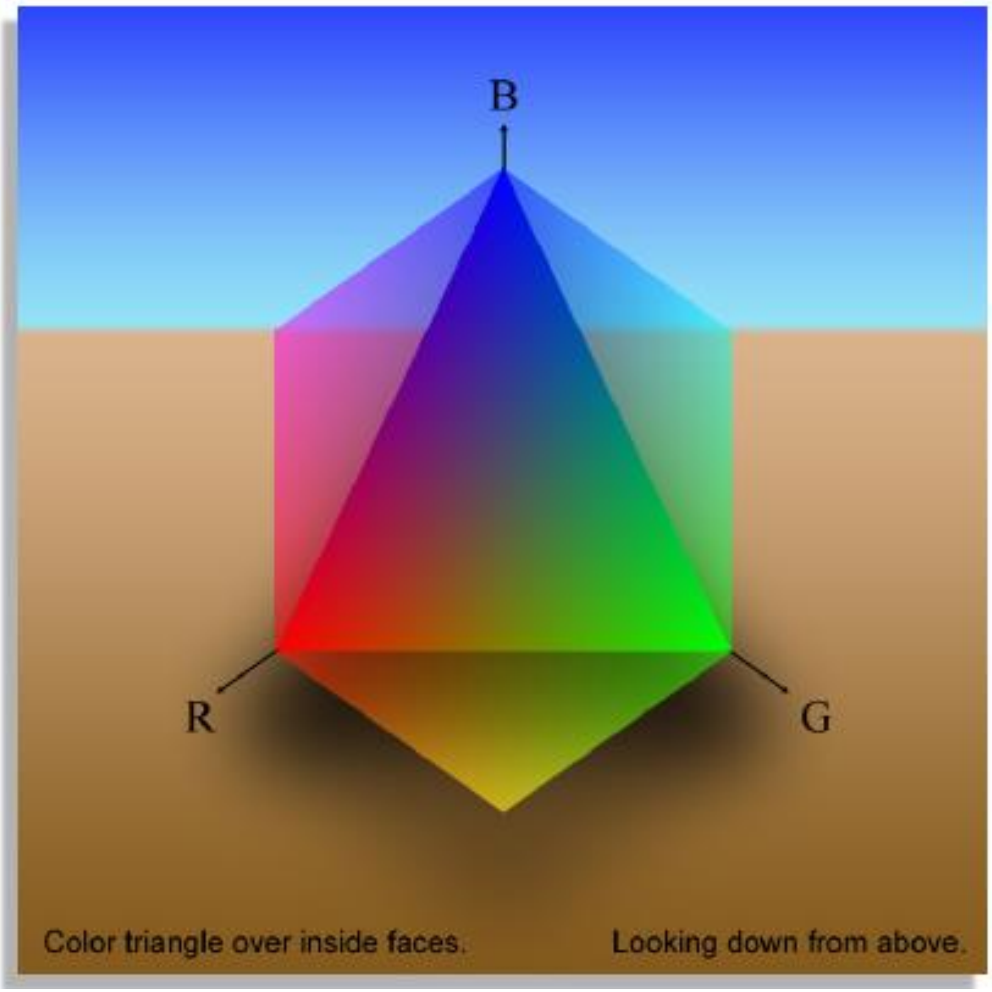
Еквівалентний кольоровий трикутник

Наприклад, у площині *g* = *0*, сторона трикутника – це всі кольори, сума червоної та синьої складових яких дорівнює 255. Така сторона показана на рисунку:



Спробуйте уявити по аналогії, як це виглядає у площинах *r* = *0* та *b* = *0*.

Еквівалентний кольоровий трикутник
Як це виглядає на кольоровому кубі

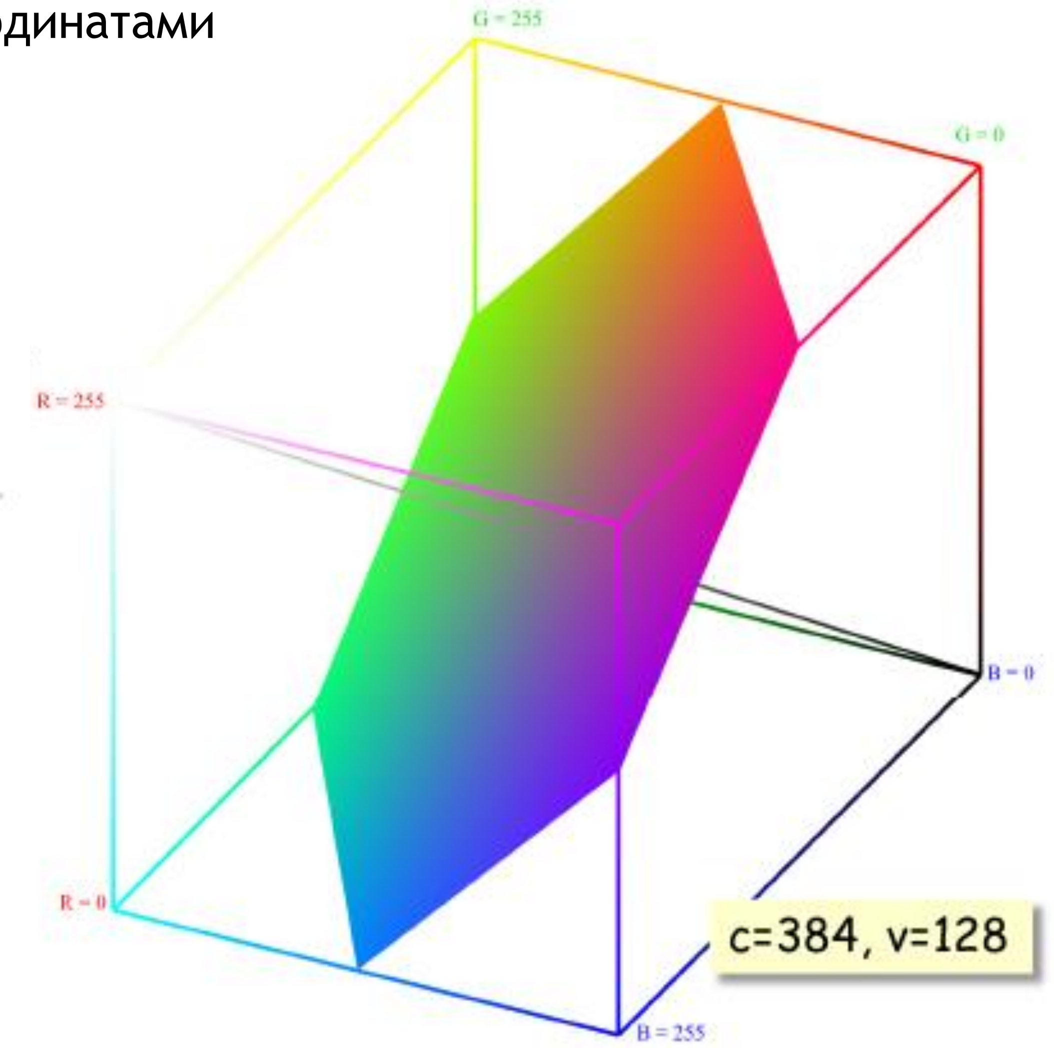


Еквівалентний кольоровий шестикутник

Якщо три вектори у просторі з координатами

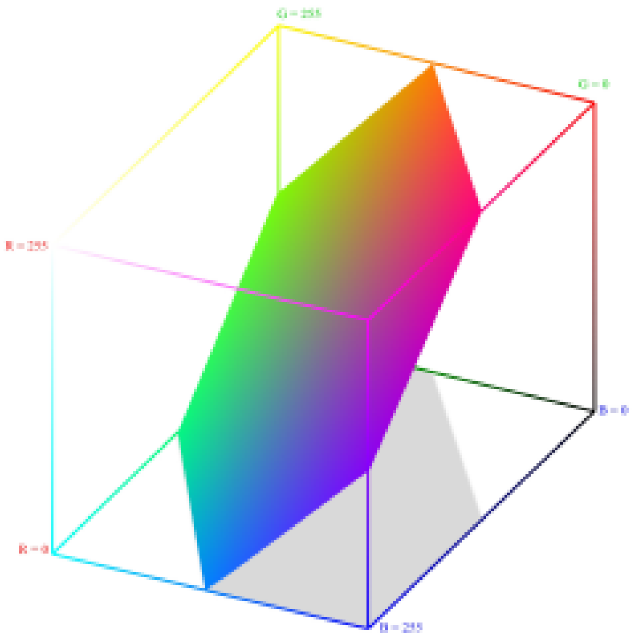
$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ c \\ 0 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix},$$

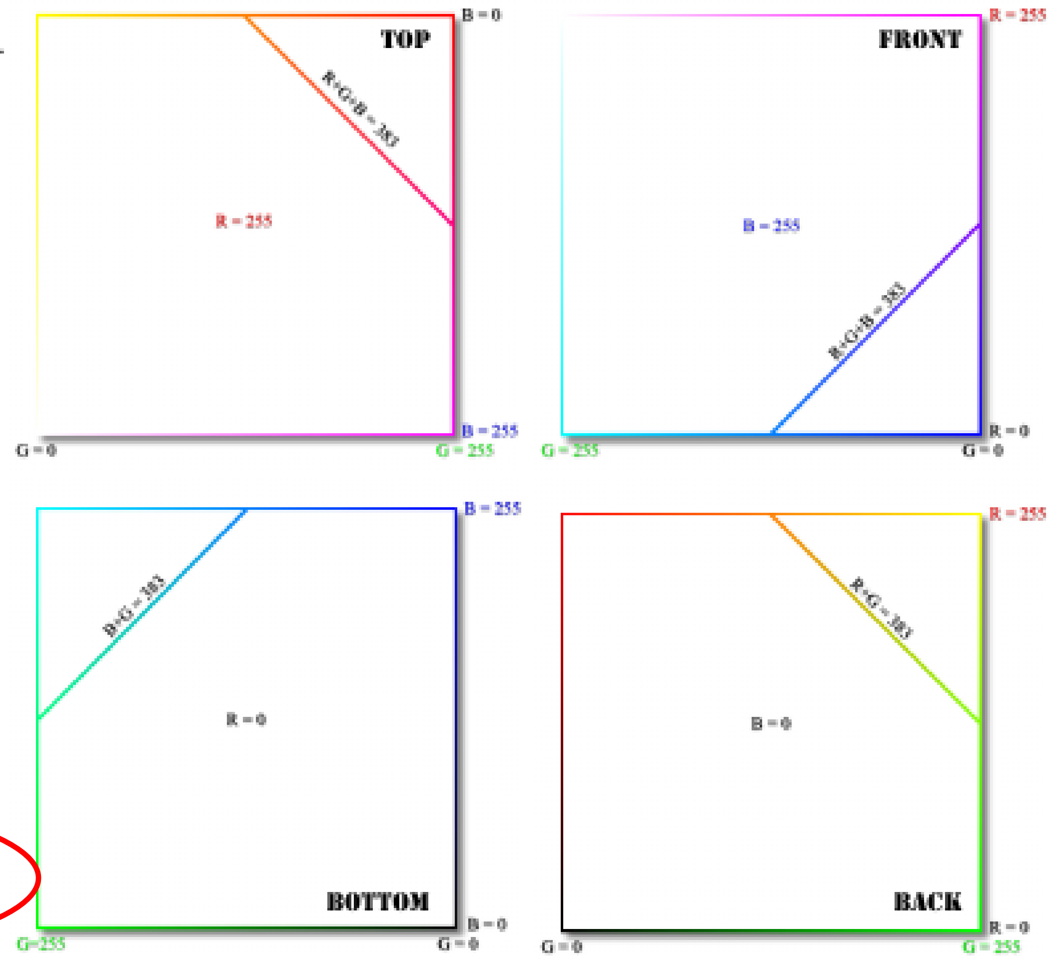
Мають сумарні середні яскравості $255 < c < 510$ ($r + g + b = c$), то вони формують так званий еквівалентний кольоровий шестикутник.



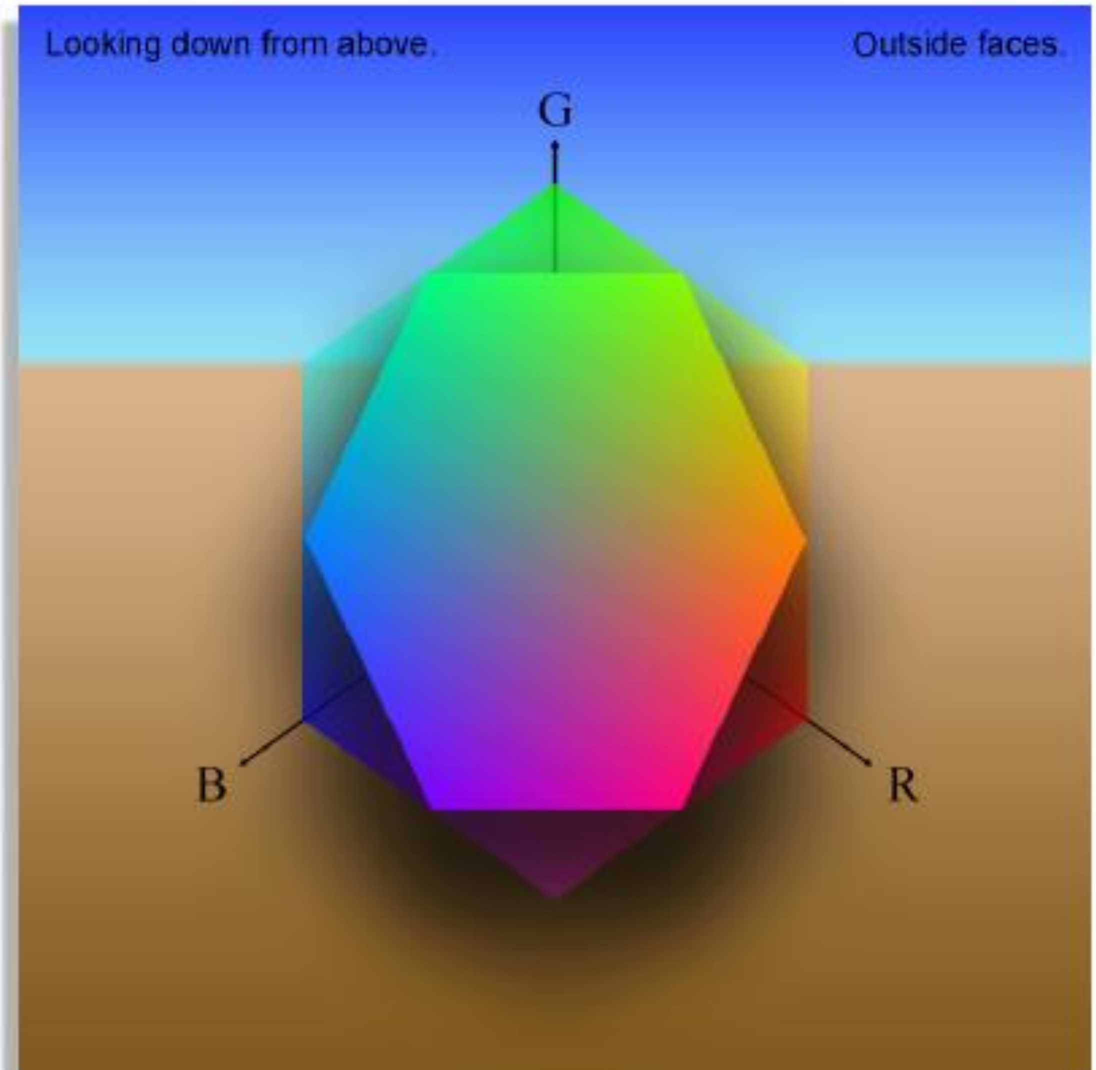
Еквівалентний кольоровий шестикутник

Outside looking in.

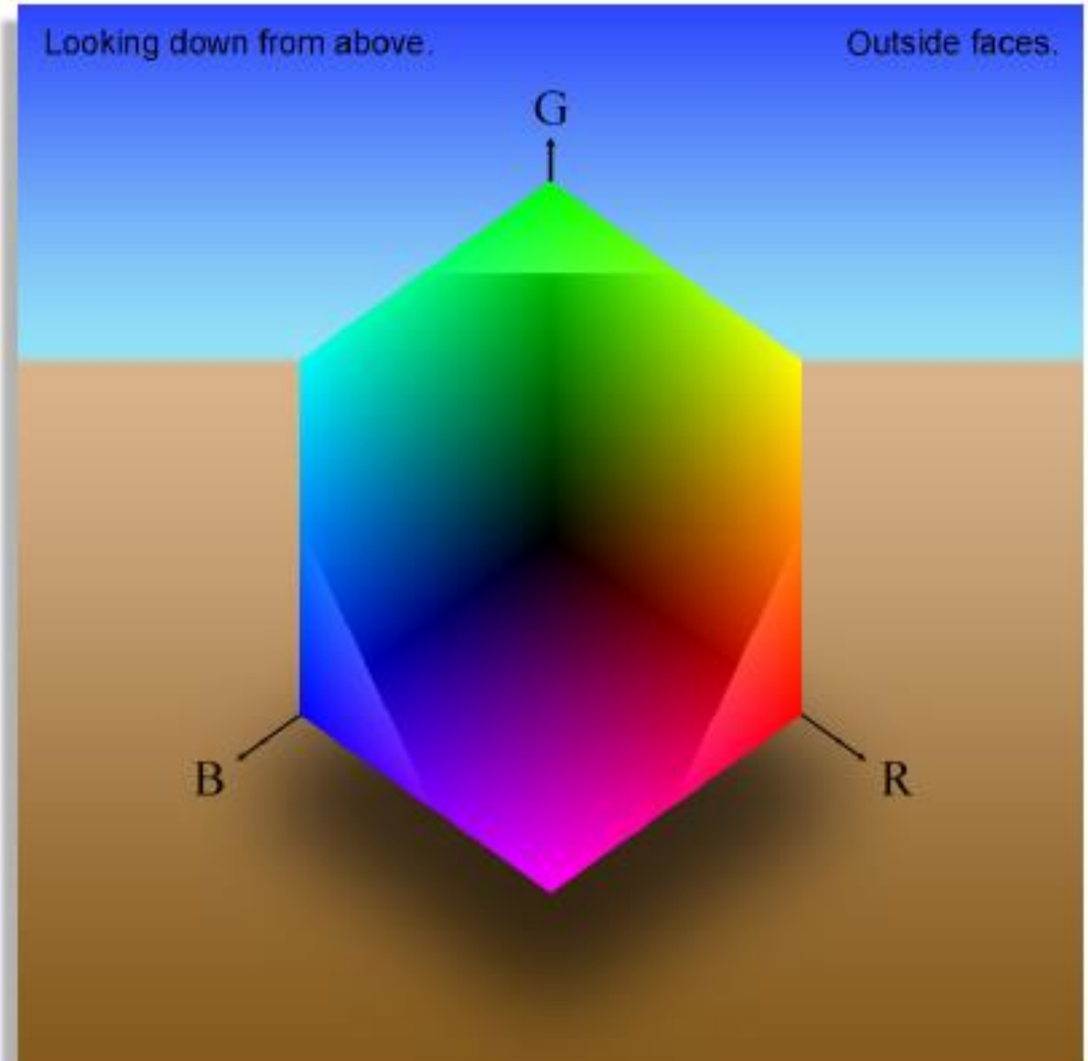




The lines on the other four faces similarly have $r + b + g = 383$.

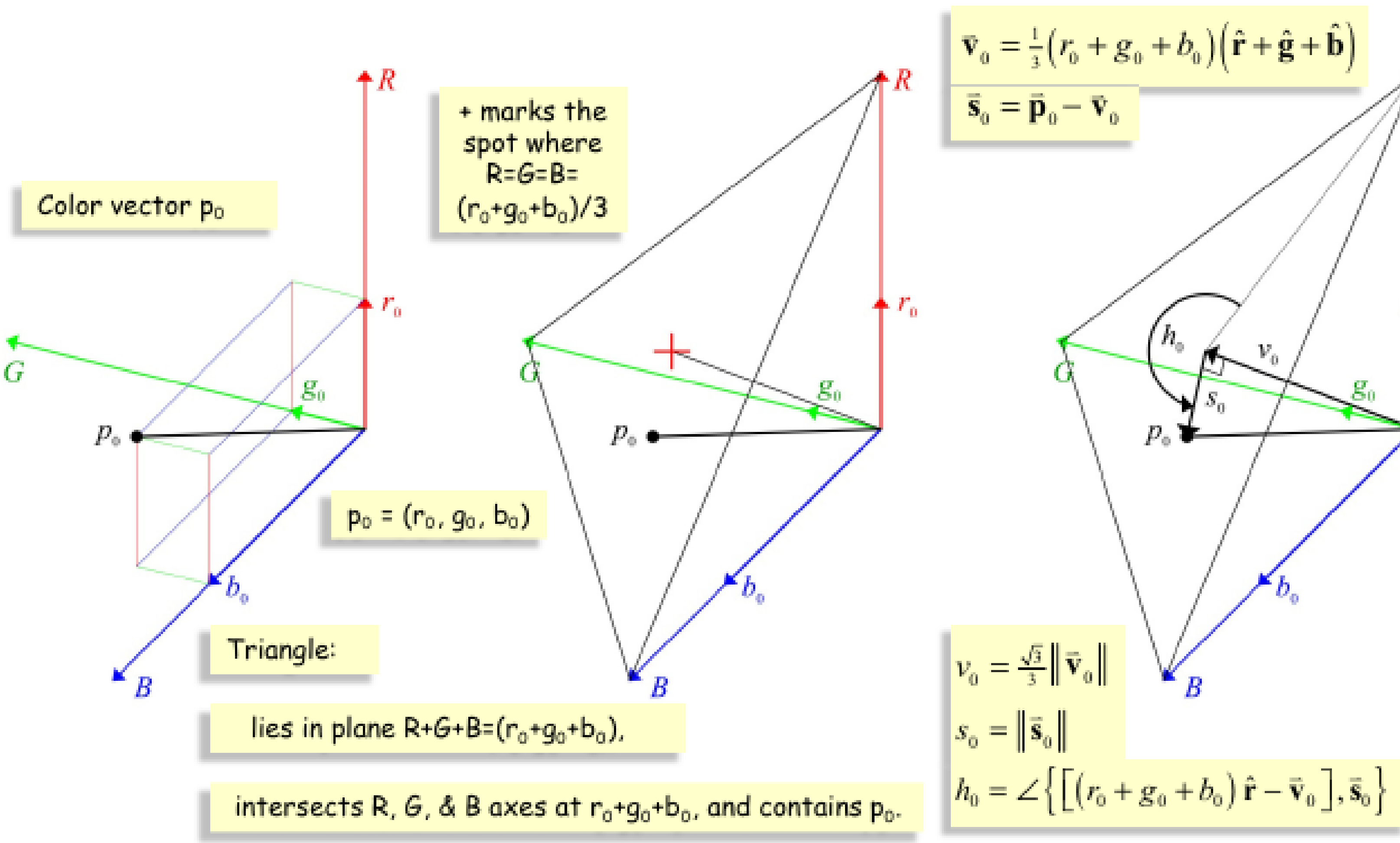

Looking down from above.
Outside faces.
G
B
R

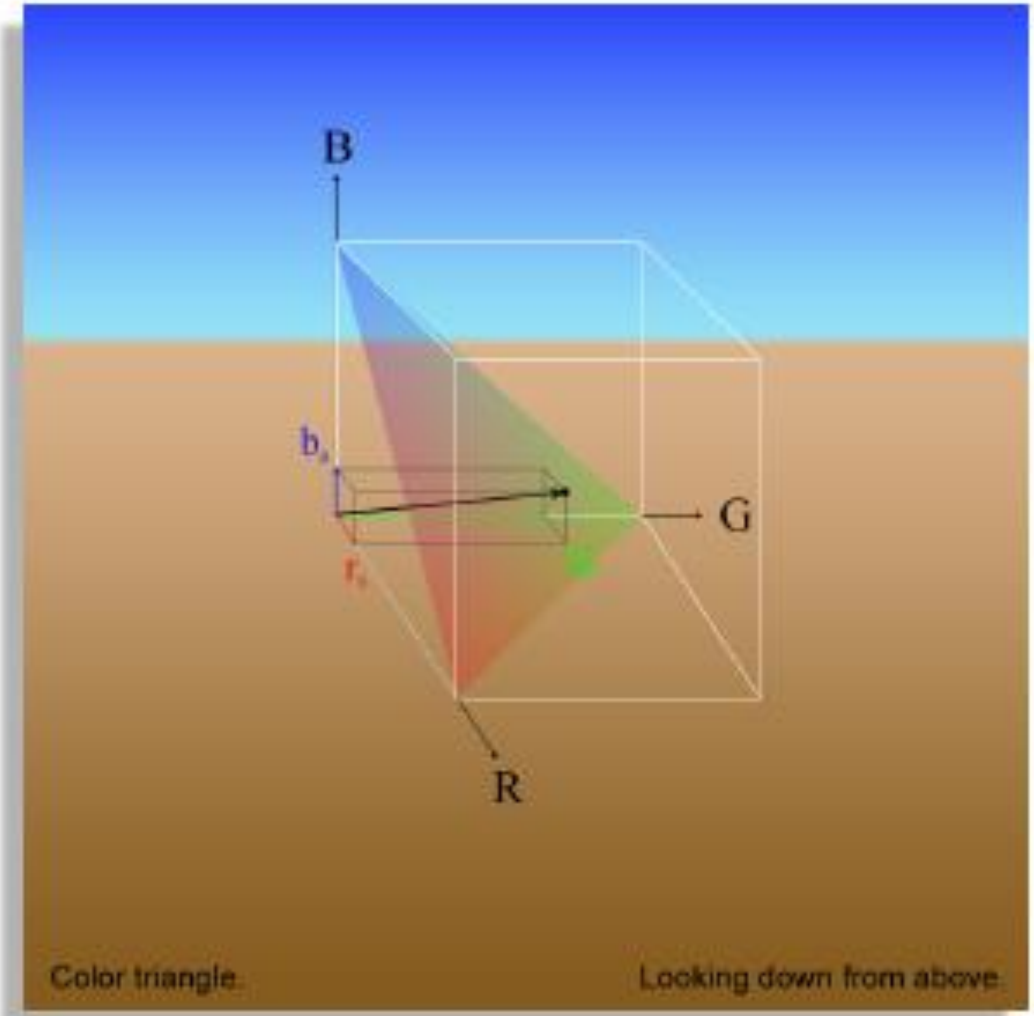
Еквівалентний кольоровий шестикутник
…і навпаки – як виглядає кольоровий куб без шестикутника



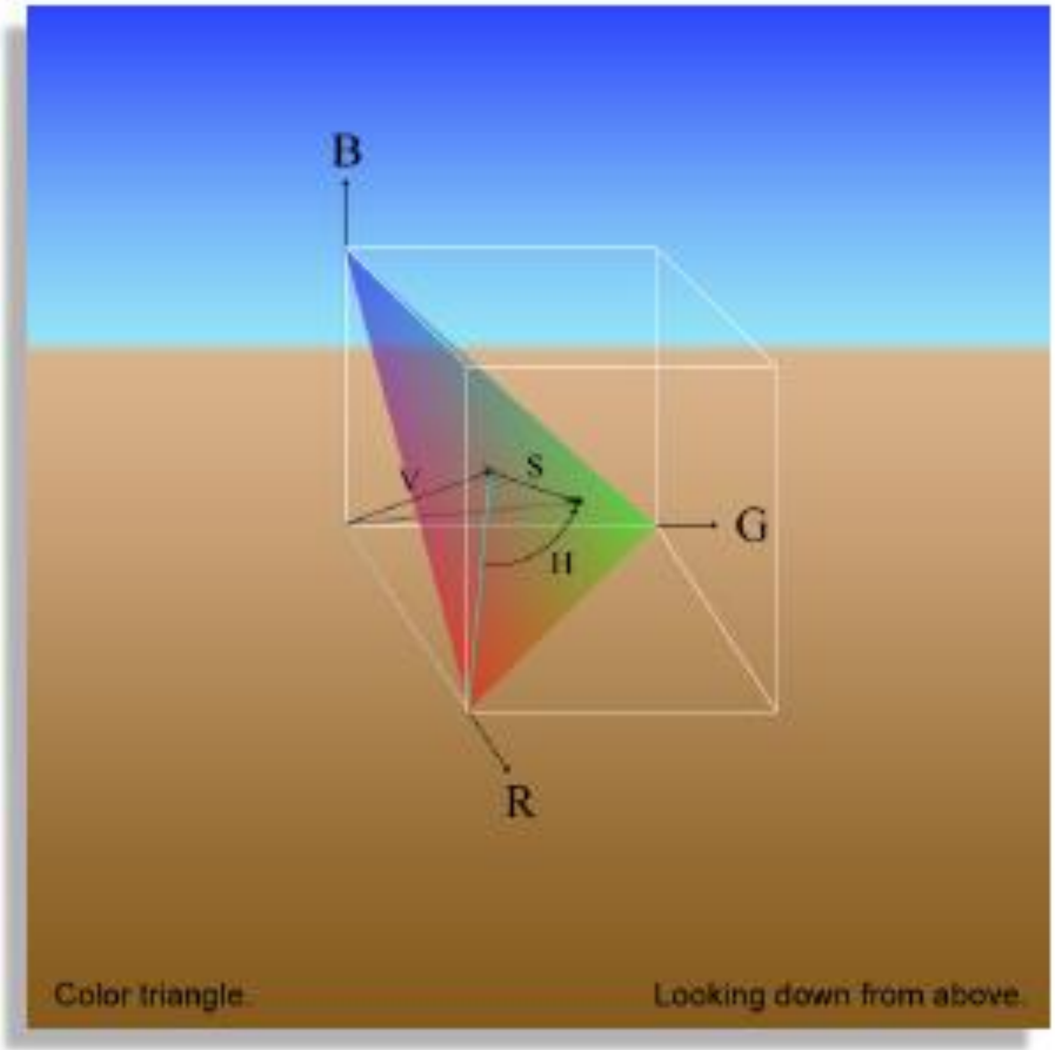
Перехід RGB => HSV

…а детальніше буде далі…

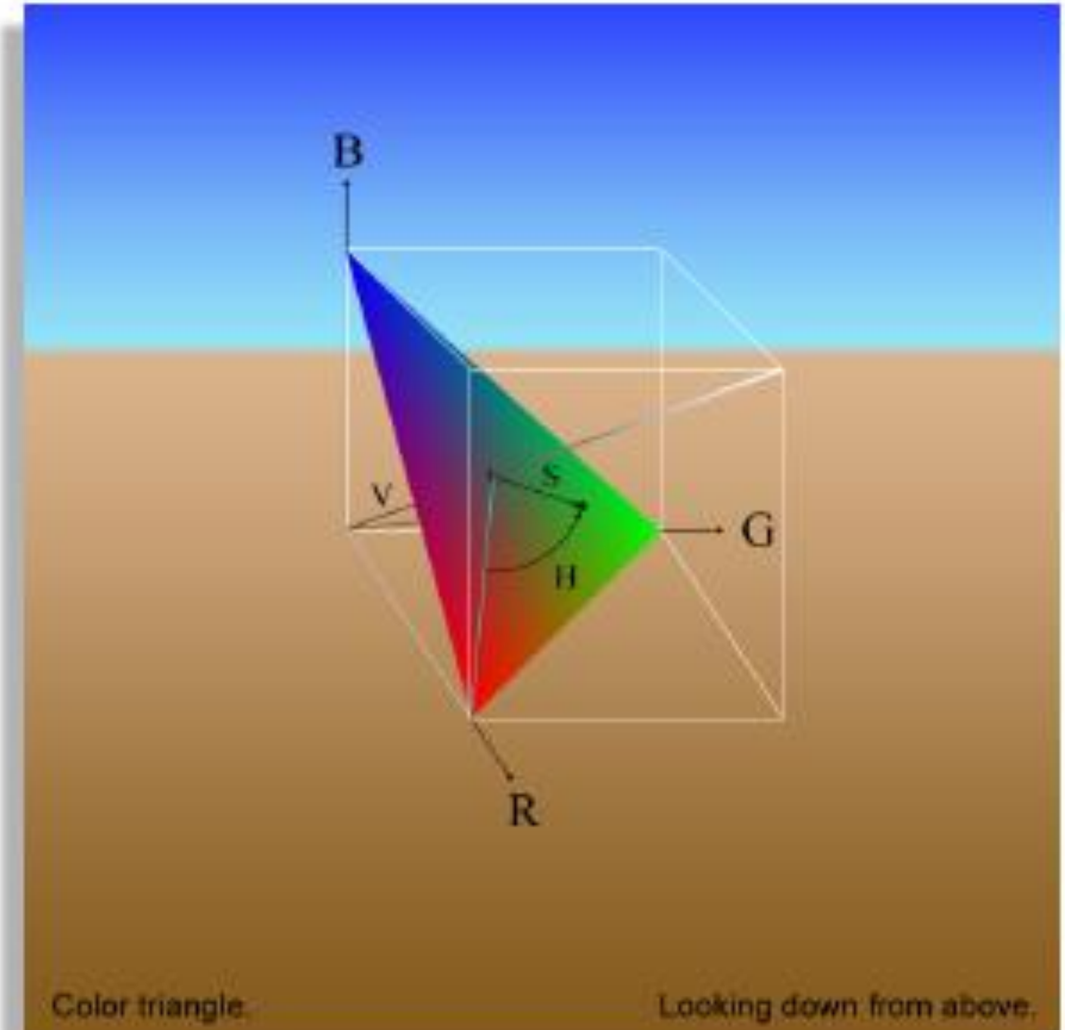



B
b
G
r
R
Color triangle
Looking down from above

Вектори яскравості, відтінку та насиченості

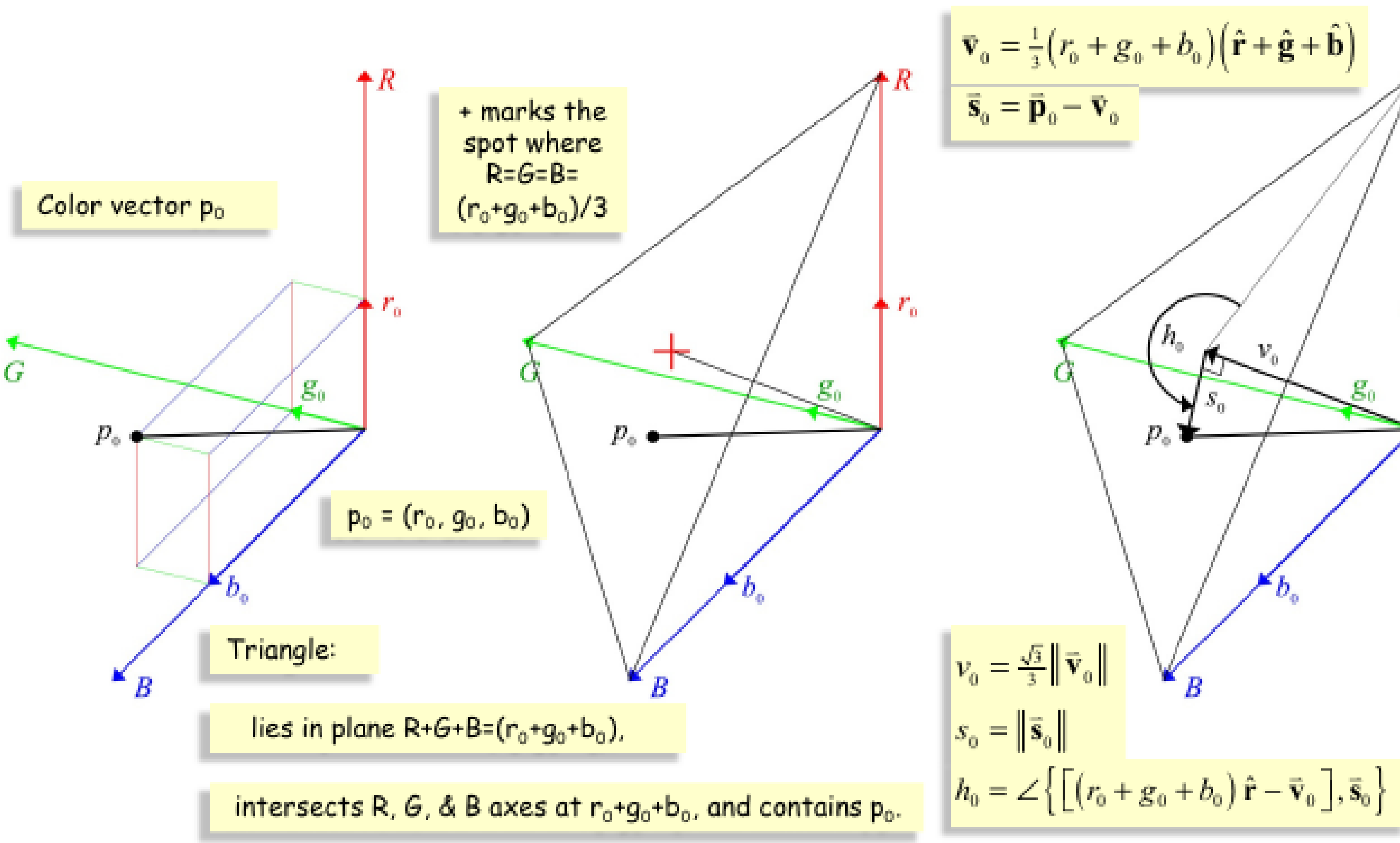


Вектори яскравості, відтінку та насиченості
Відносно сірої лінії



Перехід RGB => HSV

Ось так це виглядало в минулій лекції…



Визначаємо вектор кольору точки з координатами:

$$\mathbf{p}_0 = [r_0 \;\; g_0 \;\; b_0]^{\mathrm{T}}$$

$c = r_0 + g_0 + b_0$

Тоді яскравість точки задається вектором

$$\mathbf{v}_0 = \tfrac{1}{3}[c \;\; c \;\; c]^{\mathrm{T}}$$

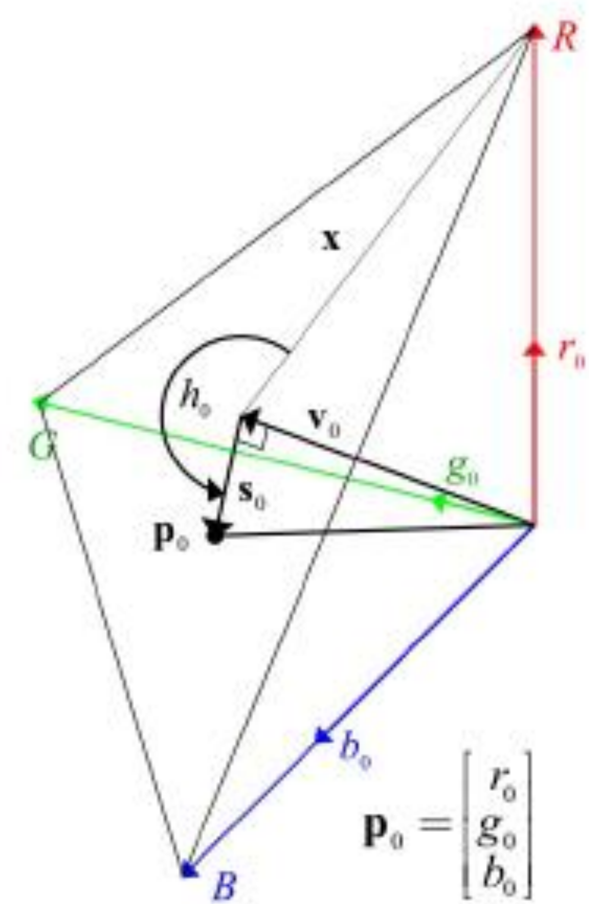
Довжина цього вектора буде дорівнювати:

$$\|\mathbf{v}_0\| = \tfrac{1}{3}\sqrt{c^2 + c^2 + c^2} = \tfrac{1}{3}\sqrt{3c^2} = \tfrac{\sqrt{3}}{3}c.$$

І, отже скалярне значення довжини цього вектора буде дорівнювати:

$$v_0 = \tfrac{1}{3}c = \tfrac{1}{\sqrt{3}}\|\mathbf{v}_0\| = \tfrac{\sqrt{3}}{3}\|\mathbf{v}_0\|.$$

Отримане значення – це складова *V* (яскравість) у колірному просторі HSV.



$c = r_0 + g_0 + b_0$

$$\mathbf{x} = \mathbf{R} - \mathbf{v}_0 = \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix} - \tfrac{1}{3}\begin{bmatrix} c \\ c \\ c \end{bmatrix} = \frac{c}{3}\begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix},$$

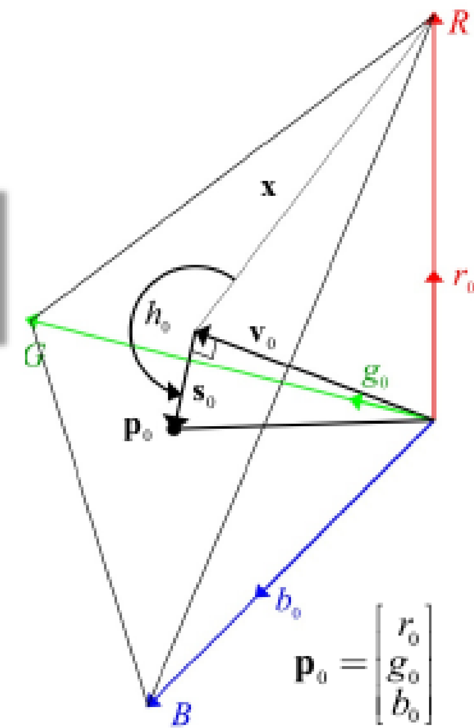
is the red line, $\mathbf{x}$, for all colors with value, $v_0 = c/3$.

$$\mathbf{s}_0 = \mathbf{p}_0 - \mathbf{v}_0 = \begin{bmatrix} r_0 - v_0 \\ g_0 - v_0 \\ b_0 - v_0 \end{bmatrix},$$

is the saturation vector, $\mathbf{s}_0$, for color $\mathbf{p}_0$.

$$h_0 = \angle(\mathbf{s}_0, \mathbf{x}) = \cos^{-1}\left(\frac{\mathbf{s}_0}{\|\mathbf{s}_0\|} \bullet \frac{\mathbf{x}}{\|\mathbf{x}\|}\right),$$

is the hue angle, h_0, for color $\mathbf{p}_0$.



$\mathbf{p}_0 = \begin{bmatrix} r_0 \\ g_0 \\ b_0 \end{bmatrix}$

Перехід від RGB до HSV
Підсумкові формули для *V* та *S*

$c = r_0 + g_0 + b_0$

$\mathbf{p}_0$ is the color **vector**

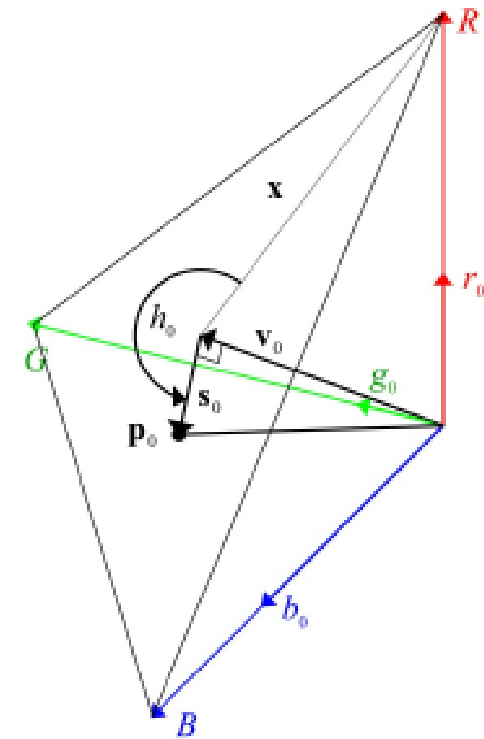
$$\mathbf{p}_0 = \begin{bmatrix} r_0 \\ g_0 \\ b_0 \end{bmatrix}.$$

$\mathbf{v}_0$ is the value **vector**.

$$\mathbf{v}_0 = \tfrac{1}{3}\begin{bmatrix} c \\ c \\ c \end{bmatrix}, \quad c = r_0 + g_0 + b_0.$$

v_0 is the **value** — a scalar.

$$v_0 = \tfrac{1}{3}c, \quad \left\|\mathbf{v}_0\right\| = \frac{\sqrt{3}}{3}c \Rightarrow v_0 = \frac{\sqrt{3}}{3}\left\|\mathbf{v}_0\right\|.$$



$\mathbf{s}_0$ is the saturation **vector**

$$\mathbf{s}_0 = \mathbf{p}_0 - \mathbf{v}_0 = \begin{bmatrix} r_0 - v_0 \\ g_0 - v_0 \\ b_0 - v_0 \end{bmatrix}.$$

s_0 is the **saturation** — a scalar.

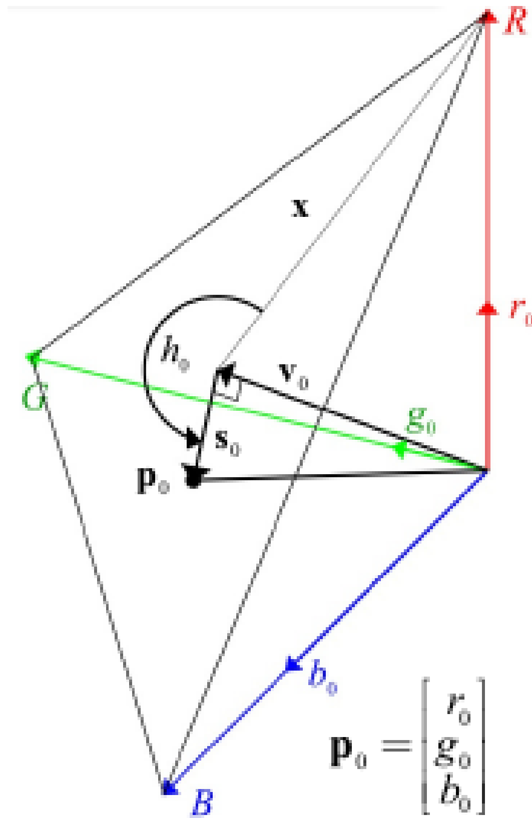
$$s_0 = \left\|\mathbf{s}_0\right\| = \sqrt{(r_0 - v_0)^2 + (g_0 - v_0)^2 + (b_0 - v_0)^2}.$$

Нормалізована насиченість

Скалярне значення насиченості

$$s_0 = \sqrt{(r_0 - v_0)^2 + (g_0 - v_0)^2 + (b_0 - v_0)^2},$$

як правило, є дробовим числом, що для багатьох задач є незручним. Краще, щоб насиченість була *нормалізованою*, тобто могла приймати значення лише від 0 до 1. Тоді у насиченості з'являється ясний зміст: 0 – це відсутність певного кольору, а 1 – його максимальна насиченість.
Нормалізувати насиченість найпростіше наступним способом. Знайдемо найдовший можливий вектор насиченості в кольоровому кубі. Очевидно, що він буде лежати у лощині кольорового трикутника, вершини якого мають координати [255, 0, 0]^T, [0, 255, 0]^T, та [0, 0, 255]^T.
Від заданої точки $p_0 = [r_0, g_0, b_0]^T$ існують три вектори до відповідних вершин трикутника. Припустимо, що найдовший – до червоної вершини, тоді:



$$\mathbf{s}_{max} = [255 \quad 0 \quad 0]^T - \tfrac{1}{3}[255 \quad 255 \quad 255]^T = [170 \quad -85 \quad -85]^T$$

$$s_{max} = \|\mathbf{s}_{max}\| \approx 208.2066.$$

Таким чином, значення s_0 може бути замінене на s_0/s_{max}.

Швидке обчислення відтінку

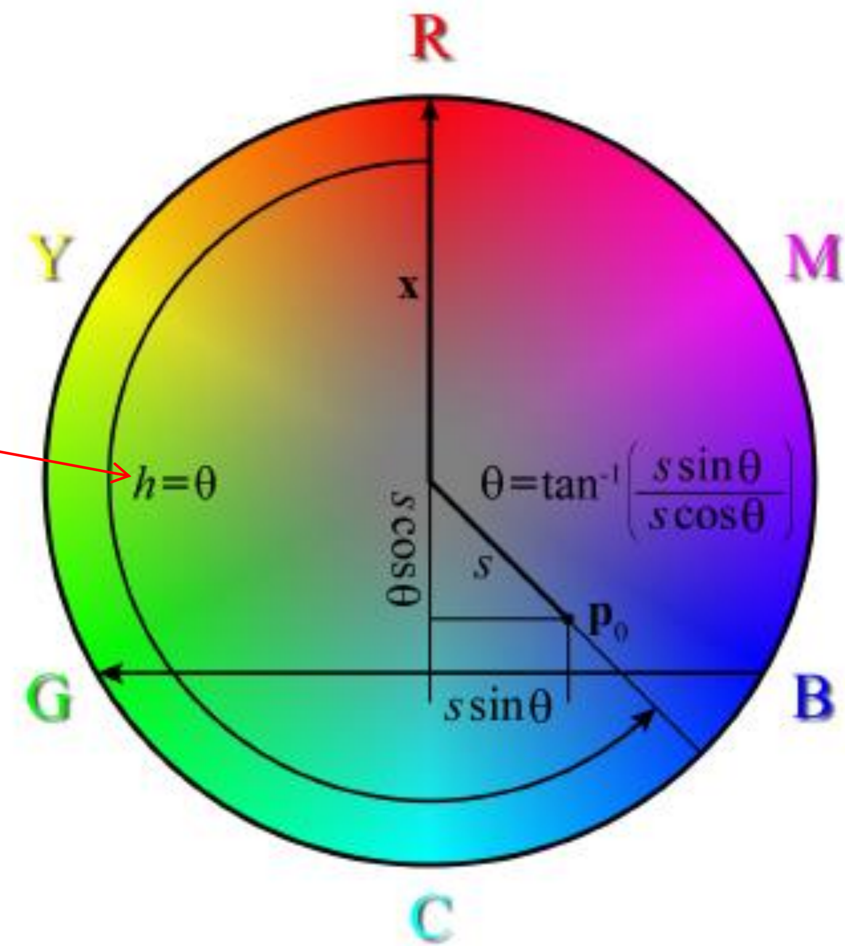
Зверніть увагу, що для обчислення відтінку треба мати значення насиченості, а для того, щоб обчислити насиченість, треба обчислити яскравість.

Існує спосіб обчислити відтінок швидше.

Відтінок – це кут на кольоровому колі

Для обчислення кута можна скористатися виразом:

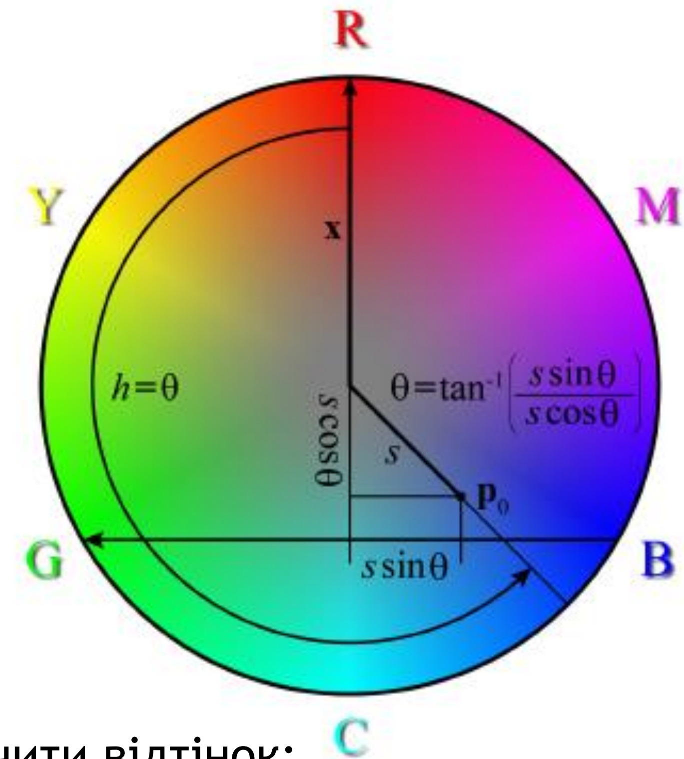
$$\arctan\left\{\frac{s\sin(\theta)}{s\cos(\theta)}\right\} = \arctan\left\{\frac{\sin(\theta)}{\cos(\theta)}\right\} = \theta$$



Швидке обчислення відтінку

Проекція вектора $\mathbf{p}_0$ на лінію червоного кольору $\mathbf{x}$:

$$\mathbf{x}\bullet\mathbf{p} = \frac{c}{3}\begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix}\bullet\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \frac{c}{3}[2r-g-b].$$



Вектор $\mathbf{z}=\sqrt{3}\frac{c}{3}[0 \quad 1 \quad -1]^{\mathrm{T}}$ є перпендикулярним до $\mathbf{x}$.
Проекція $\mathbf{p}$ на $\mathbf{z}$:

$$\mathbf{z}\bullet\mathbf{p} = \frac{c\sqrt{3}}{3}\begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}\bullet\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \frac{c\sqrt{3}}{3}[g-b].$$

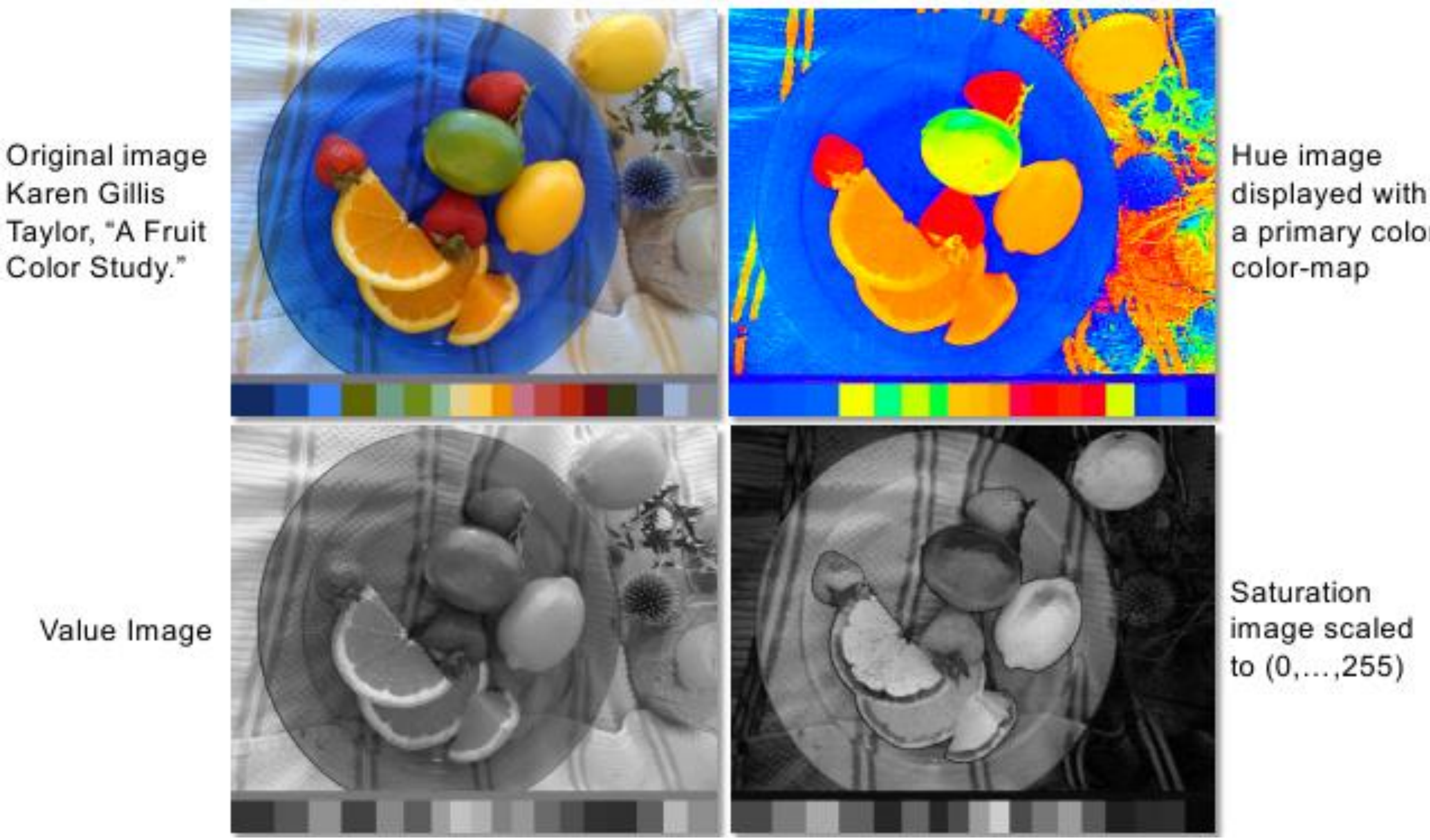
Відношення цих двох величин дає можливість визначити відтінок:

$$h = \tan^{-1}\left(\frac{\mathbf{z}\bullet\mathbf{p}}{\mathbf{x}\bullet\mathbf{p}}\right) = \tan^{-1}\left(\frac{\sqrt{3}(g-b)}{2r-g-b}\right).$$

Зверніть увагу, що в даному випадку відтінок обчислюється незалежно від яскравості та насиченості.

Як виглядають HSV-канали

Класичний приклад



Що дає представлення зображень в кодуванні HSV?

Зображення перетворюють з формату кодування RGB на формат кодування HSV з метою спрощення аналізу, насамперед – сегментації.

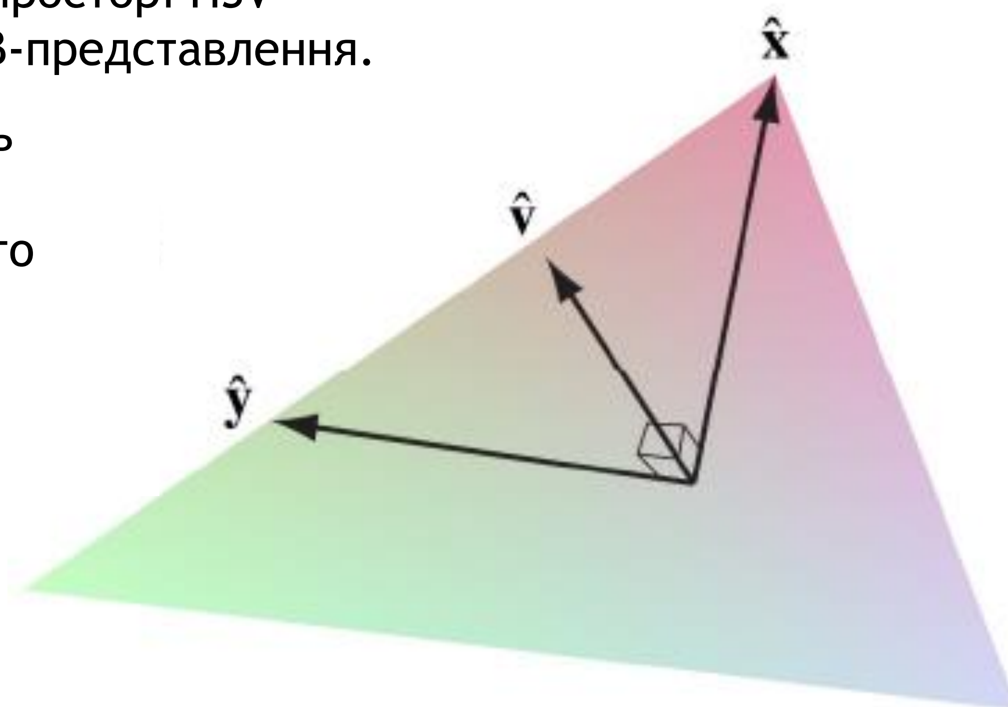
Справа в тому, що фізичні принципи реєстрації зображень у більшості випадків дають RGB – саме під цей тип кодування спроектовані та розраховані фотоматриці. Крім того, цей спосіб кодування є найбільш “природним”. Проте для сегментації (тобто виділення певної області зображення за деякою ознакою) кодування в RGB-форматі стикається зі значними алгоритмічними труднощами.

Цих недоліків позбавлено кодування у форматі HSV. У більшості випадків однорідні області зображень мають або один колір, або однакову яскравість. Таким чином, для виділення певної області достатньо опрацювати один канал, а не три, що практично трикратний виграш у швидкодії алгоритмів сегментації.

Перехід HSV => RGB

Після всіх перетворень у колірному просторі HSV виникає потреба повернутися до RGB-представлення.

Перетворення RGB -> HSV переводить колірний простір у еквівалентний кольоровий трикутник, площина якого перпендикулярна до яскравості (вектора **v**).



Таким чином, у площині еквівалентного кольорового трикутника можна виділити перпендикулярні вектори **x** та **y**, такі, що разом з **v** будуть утворювати базис.

Якщо ми врахуємо лише напрямок (без величини (модуля)) цих векторів, то їх можна записати як одиничні вектори:

$$\hat{\mathbf{x}} = \begin{bmatrix}1\\0\\0\end{bmatrix}, \quad \hat{\mathbf{y}} = \begin{bmatrix}0\\1\\0\end{bmatrix}, \quad \hat{\mathbf{v}} = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

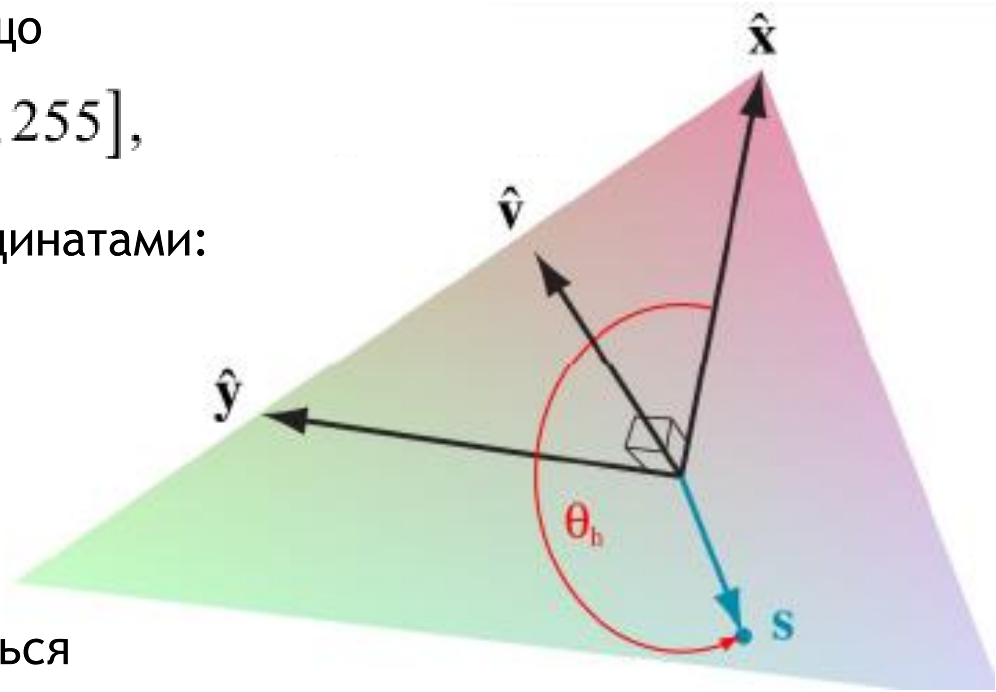
Перехід HSV => RGB

Якщо у нас є значення *h*, *s* та *v*, такі, що

$$h \in [0, 2\pi), \quad s \in [0, s_{\max}], \quad v \in [0, 255],$$

то вектор насиченості задається координатами:

$$[\mathbf{s}]_{\mathrm{xyv}} = \begin{bmatrix} s\cos(h) \\ s\sin(h) \\ 0 \end{bmatrix}_{\mathrm{xyv}},$$



а через вектори **x**, **y** та **v** він виражається наступним чином:

$$\mathbf{s} = s\cos(h)\,\hat{\mathbf{x}} + s\sin(h)\,\hat{\mathbf{y}} + 0\,\hat{\mathbf{v}}.$$

Легко бачити, що це вектор, повернутий відносно вектора **x** на деякий кут θ_b.

Перехід HSV => RGB

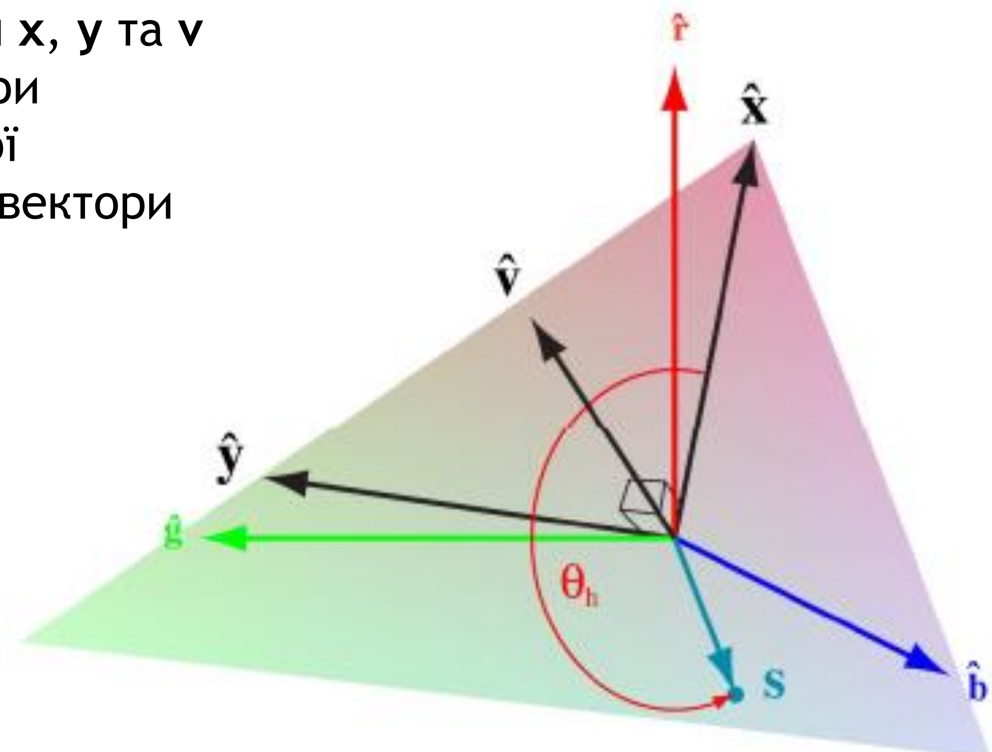
Важливо відзначити, що хоча вектори **x**, **y** та **v** утворюють базис, так само як і вектори відповідно червоної, зеленої та синьої складових кольору, але це не ті самі вектори **r**, **g** та **b**.

Таким чином, $[\mathbf{s}]_{xyv}$, яке ми знаємо, не дорівнює $[\mathbf{s}]_{rgb}$, яке ми не знаємо. Але ми можемо знайти вектор кольору **p** і розкласти його по базису RGB і таким чином знайти складові кольору.

$$[\mathbf{s}]_{\mathbf{rgb}} = [r \quad g \quad b]^{\mathsf{T}}$$

$$\mathbf{s} \leftrightarrow r\hat{\mathbf{r}} + g\hat{\mathbf{g}} + b\hat{\mathbf{b}},$$

$$\mathbf{s} \leftrightarrow s\cos(h)\hat{\mathbf{x}} + s\sin(h)\hat{\mathbf{y}} + 0\,\hat{\mathbf{v}}.$$



$$[\mathbf{s}]_{\mathbf{rgb}} = [r \quad g \quad b]^{\mathsf{T}}$$

$$[\mathbf{s}]_{\mathbf{xyz}} = \left[s\cos(h) \quad s\sin(h) \quad 0\right]^{\mathsf{T}}$$

$$[\mathbf{s}]_{\mathbf{rgb}} \neq [\mathbf{s}]_{\mathbf{xyz}}$$

Перехід HSV => RGB

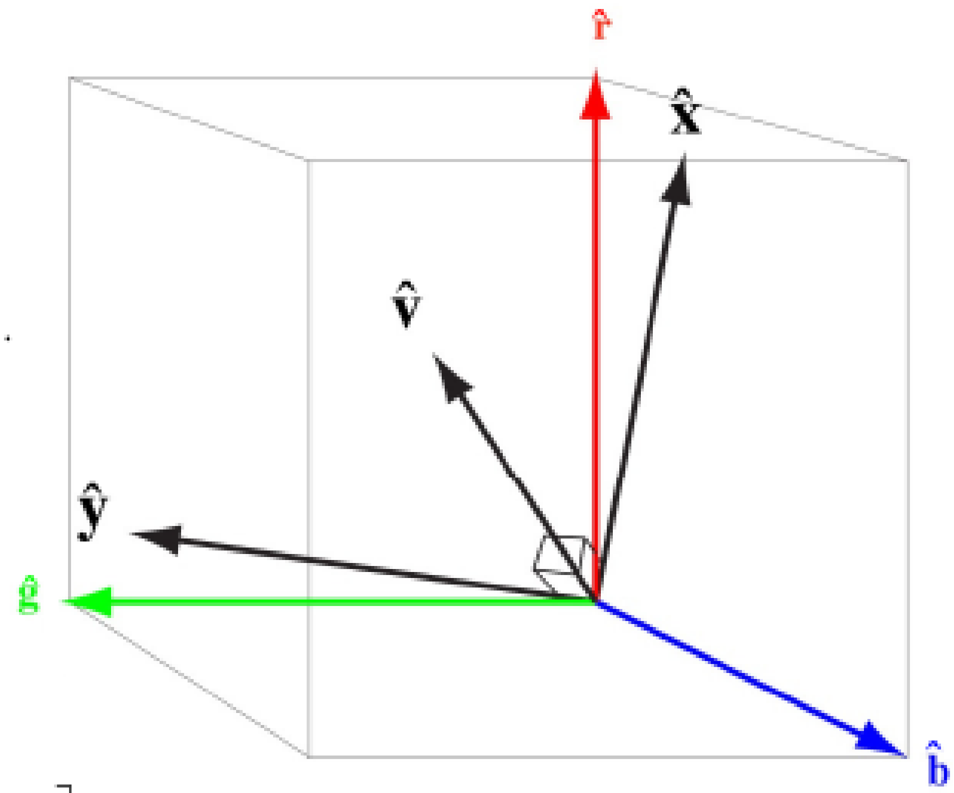
Математично таке перетворення описується так званою *матрицею повороту*, тобто такою матрицею, що її добуток зліва на вектор призводить до повороту вектора:

$$[\hat{\mathbf{x}}]_{\mathrm{rgb}} = A[\hat{\mathbf{x}}]_{\mathrm{xyv}}, \ [\hat{\mathbf{y}}]_{\mathrm{rgb}} = A[\hat{\mathbf{y}}]_{\mathrm{xyv}}, \ [\hat{\mathbf{v}}]_{\mathrm{rgb}} = A[\hat{\mathbf{v}}]_{\mathrm{xyv}}.$$

У трьохвимірному простору це квадратна матриця 3x3.

Останню формулу можна розписати так:

$$\begin{aligned}
[\mathbf{s}]_{\mathrm{rgb}} &= A[\mathbf{s}]_{\mathrm{xyv}} \\
&= A\left[s\cos(h)[\hat{\mathbf{x}}]_{\mathrm{xyv}} + s\sin(h)[\hat{\mathbf{y}}]_{\mathrm{xyv}} + 0[\hat{\mathbf{v}}]_{\mathrm{xyv}} \right] \\
&= s\cos(h)\,A[\hat{\mathbf{x}}]_{\mathrm{xyv}} + s\sin(h)\,A[\hat{\mathbf{y}}]_{\mathrm{xyv}} + 0\,A[\hat{\mathbf{v}}]_{\mathrm{xyv}} \\
&= s\cos(h)[\hat{\mathbf{x}}]_{\mathrm{rgb}} + s\sin(h)[\hat{\mathbf{y}}]_{\mathrm{rgb}} + 0[\hat{\mathbf{v}}]_{\mathrm{rgb}}.
\end{aligned}$$



Перехід HSV => RGB

Вектори початкового базису мають координати:

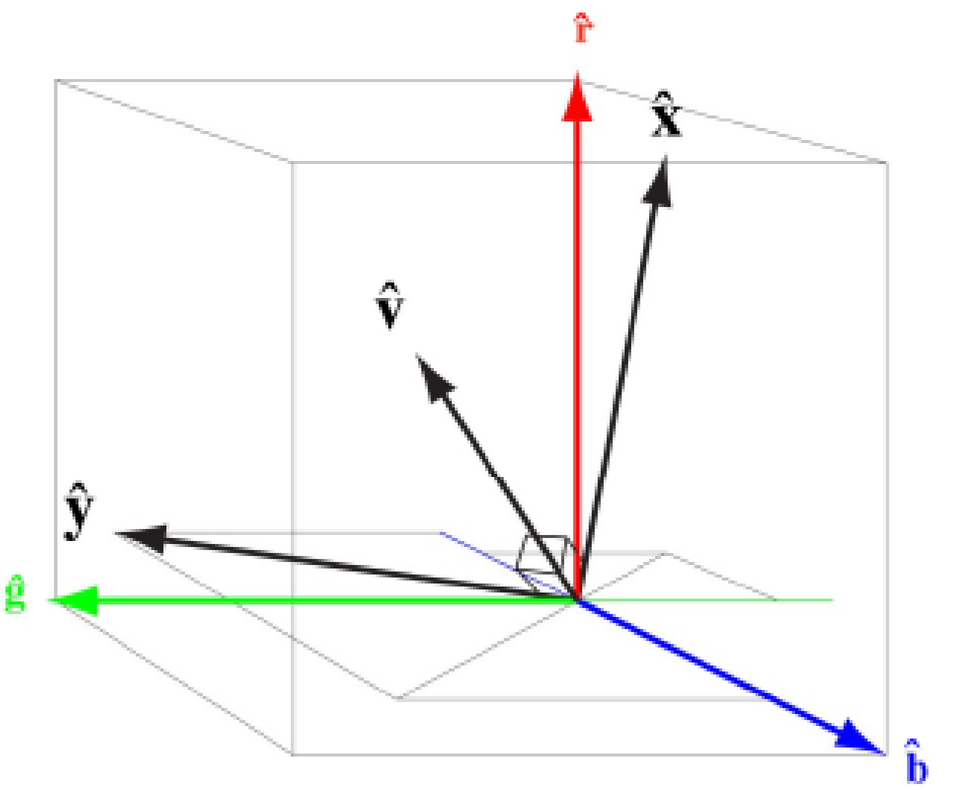
$$\hat{\mathbf{x}} = \begin{bmatrix}1\\0\\0\end{bmatrix},\ \hat{\mathbf{y}} = \begin{bmatrix}0\\1\\0\end{bmatrix},\ \hat{\mathbf{v}} = \begin{bmatrix}0\\0\\1\end{bmatrix}$$

Тоді вектори кінцевого базису матимуть координати:

$$[\hat{\mathbf{x}}]_{\mathbf{rgb}} = A\begin{bmatrix}1\\0\\0\end{bmatrix},\ [\hat{\mathbf{y}}]_{\mathbf{rgb}} = A\begin{bmatrix}0\\1\\0\end{bmatrix},\ [\hat{\mathbf{v}}]_{\mathbf{rgb}} = A\begin{bmatrix}0\\0\\1\end{bmatrix}.$$

І якщо звести воєдино ці вирази, то матрицю повороту можна виразити так:

$$A = \begin{bmatrix} [\hat{\mathbf{x}}]_{\mathbf{rgb}} & [\hat{\mathbf{y}}]_{\mathbf{rgb}} & [\hat{\mathbf{v}}]_{\mathbf{rgb}} \end{bmatrix}.$$



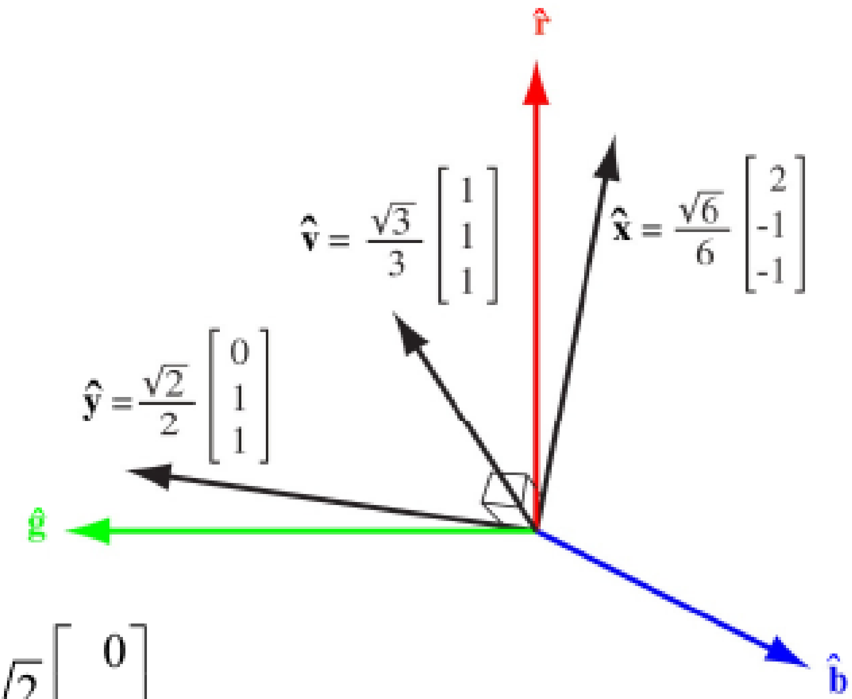
Перехід HSV => RGB

v – це одиничний вектор, записаний в *rgb*-координатах:

$$[\hat{\mathbf{v}}]_{\mathbf{rgb}} = \frac{\sqrt{3}}{3}\begin{bmatrix}1\\1\\1\end{bmatrix}$$

Вектор **x** перпендикулярний до **v** і має однакові складові *g* та *b*:

$$[\hat{\mathbf{x}}]_{\mathbf{rgb}} = \frac{\sqrt{6}}{6}\begin{bmatrix}2\\-1\\-1\end{bmatrix}$$



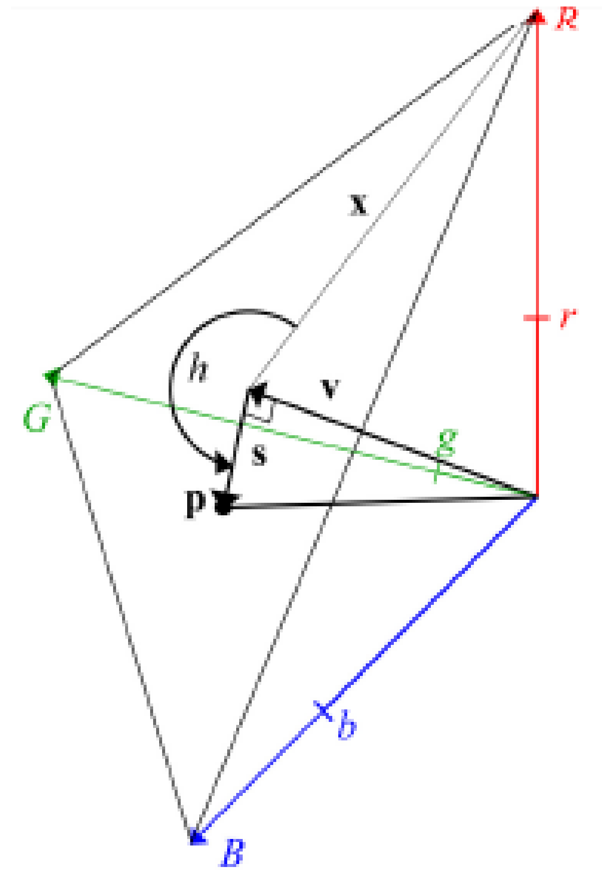
Вектор **y** є векторним добутком **x** та **v**:

$$[\hat{\mathbf{y}}]_{\mathbf{rgb}} = [\hat{\mathbf{v}}]_{\mathbf{rgb}} \times [\hat{\mathbf{x}}]_{\mathbf{rgb}} = \frac{\sqrt{2}}{2}\begin{bmatrix}0\\1\\-1\end{bmatrix}$$

Оскільки ці вектори утворюють базис, то вони всі одиничні, і тому перед матрицею-стовпчиком скрізь стоять коефіцієнти, які і роблять цю одиничність (так звані *нормуючі (або нормувальні) коефіцієнти*).

Таким чином, матриця повороту має вигляд:

$$A = \frac{\sqrt{6}}{6}\begin{bmatrix} 2 & 0 & \sqrt{2} \\ -1 & \sqrt{3} & \sqrt{2} \\ -1 & -\sqrt{3} & \sqrt{2} \end{bmatrix}.$$



Підставляючи це у другу формулу на слайді 5, можна отримати:

$$[\mathbf{s}]_{\mathbf{rgb}} = s\frac{\sqrt{6}}{6}\cos(h)\begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix} + s\frac{\sqrt{2}}{2}\sin(h)\begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} + 0\frac{\sqrt{3}}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= s\frac{\sqrt{6}}{6}\cos(h)\begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix} + s\frac{\sqrt{2}}{2}\sin(h)\begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}.$$

Перехід HSV => RGB

І, нарешті, можна отримати вектор кольору:

$$\mathbf{p} = [\mathbf{p}]_{rgb} = [\mathbf{s}]_{rgb} + [\mathbf{v}]_{rgb},$$

де $[\mathbf{v}]_{rgb}$ – вектор яскравості:

$$\mathbf{v}_0 = \frac{1}{3}\begin{bmatrix} c \\ c \\ c \end{bmatrix}, \quad c = r_0 + g_0 + b_0.$$

На жаль, даний алгоритм важко назвати оптимальним, оскільки він вимагає достатньо громіздких обчислень, а на останньому етапі – розв'язку системи рівнянь для обчислення r_0, g_0 і b_0…

Тому добрі люди придумали швидкий алгоритм для переходу від HSV до RGB…

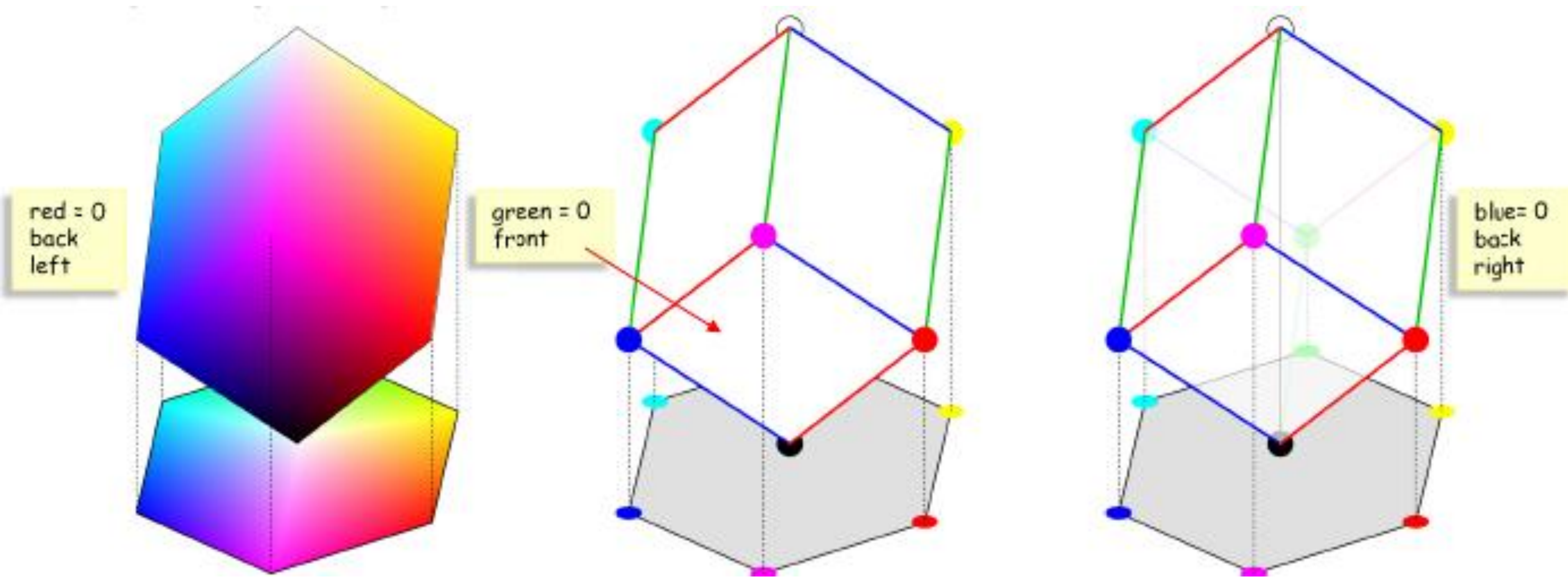
Швидкий алгоритм переходу від HSV до RGB

Важливе зауваження: будь-який "швидкий" алгоритм вимагає для своєї роботи більше пам'яті. Виграш у швидкості досягається за рахунок того, що результати проміжних обчислень зберігаються і повторно використовуються. Чим частіше вони використовуються, тим швидший алгоритм.

Для швидкого переходу від HSV до RGB використовуються проекція колірного простору HSV на RGB-кольоровий шестикутник (див. тему 7.1 "Представлення кольору. RGB та HSV" слайди "Еквівалентний кольоровий шестикутник").

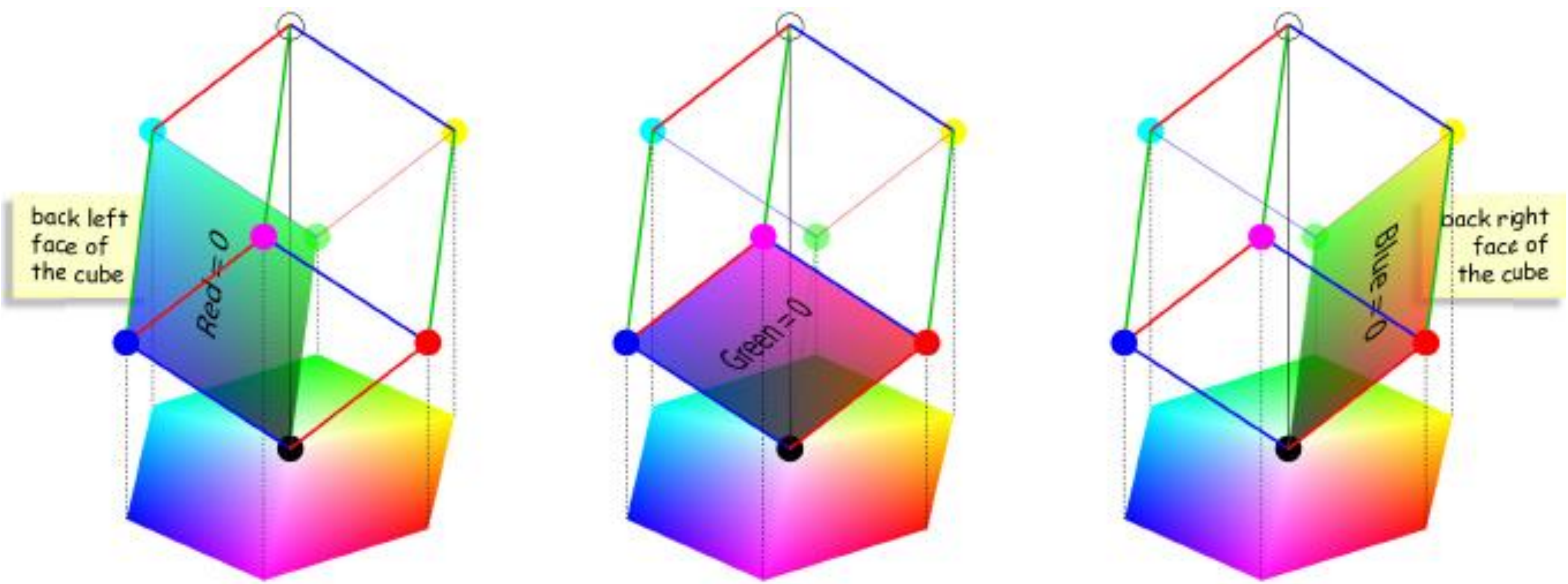
Швидкий алгоритм переходу від HSV до RGB

Нагадаємо, що еквівалентний кольоровий шестикутник получається, коли кольоровий RGB-куб проектується на площину, перпендикулярну його сірій лінії (головній діагоналі куба). В такому випадку на вершинах шестикутника знаходяться всі *первинні* (основні та додаткові) кольори: червоний, жовтий, зелений, блакитний, синій, фіолетовий.



Швидкий алгоритм переходу від HSV до RGB

Також зверніть увагу, що в такій проекції між синім та зеленим кольором відсутній червоний (r = 0), між синім і червоним – зелений (g = 0), а між червоним і зеленим – синій (b = 0).



Швидкий алгоритм переходу від HSV до RGB

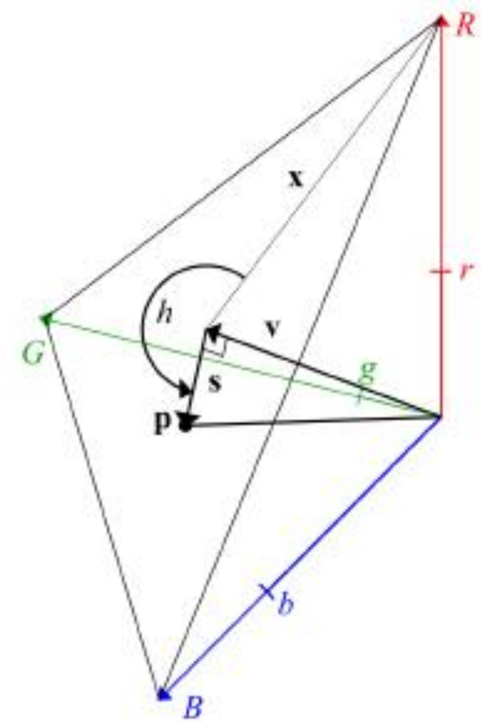
Таким чином, якщо у нас заданий вектор **h** = [h, s, v]^T , де h = [0…360), s = [0, 1] і v = [0, 1], то для обчислення вектора **p** = [r, g, b], де r, g, b = [0, 255] буде наступний алгоритм:

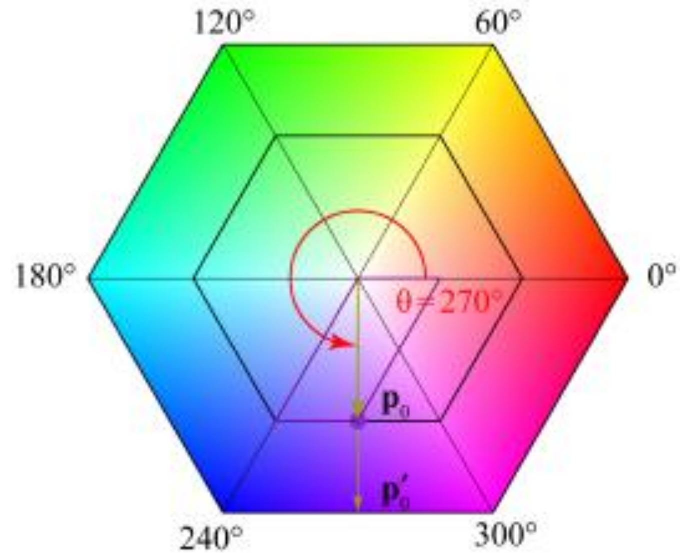
```
1. H = h/60.
2. C = v·s.
3. D = v-C.
4. X = C·(1 - |(H mod 2)-1|).
5. if        0 ≤ H < 1 then [r g b] = [C X 0]
   else if 1 ≤ H < 2 then [r g b] = [X C 0]
   else if 2 ≤ H < 3 then [r g b] = [0 C X]
   else if 3 ≤ H < 4 then [r g b] = [0 X C]
   else if 4 ≤ H < 5 then [r g b] = [X 0 C]
   else if 5 ≤ H < 6 then [r g b] = [C 0 X]
   else [r g b] = [0 0 0]
6. [r g b] = 255*[r+D g+D b+D]
```

Залишок від ділення на 2…

А весь цей блок – визначення, до якої вершини найближчий колір…

Порівняння двох алгоритмів





Другий (швидкий) алгоритм теж має векторну інтерпретацію.
Яскравість (***v***) *rgb*-точки – це проекція вектора на сіру лінію у масштабі √3/3.
Відтінок (***h***) – це кут від червоної вісі до проекції *rgb*-точки на сіру лінію.
Насиченість (***s***) – це відношення відстані від центра шестикутника до точки p_0 до відстані від центра шестикутника до будь-якої вершини.

Далі буде…

…Гістограми зображень та їх перетворення