EECE 4353 Image Processing

Lecture Notes: RGB and HSV Color Spaces

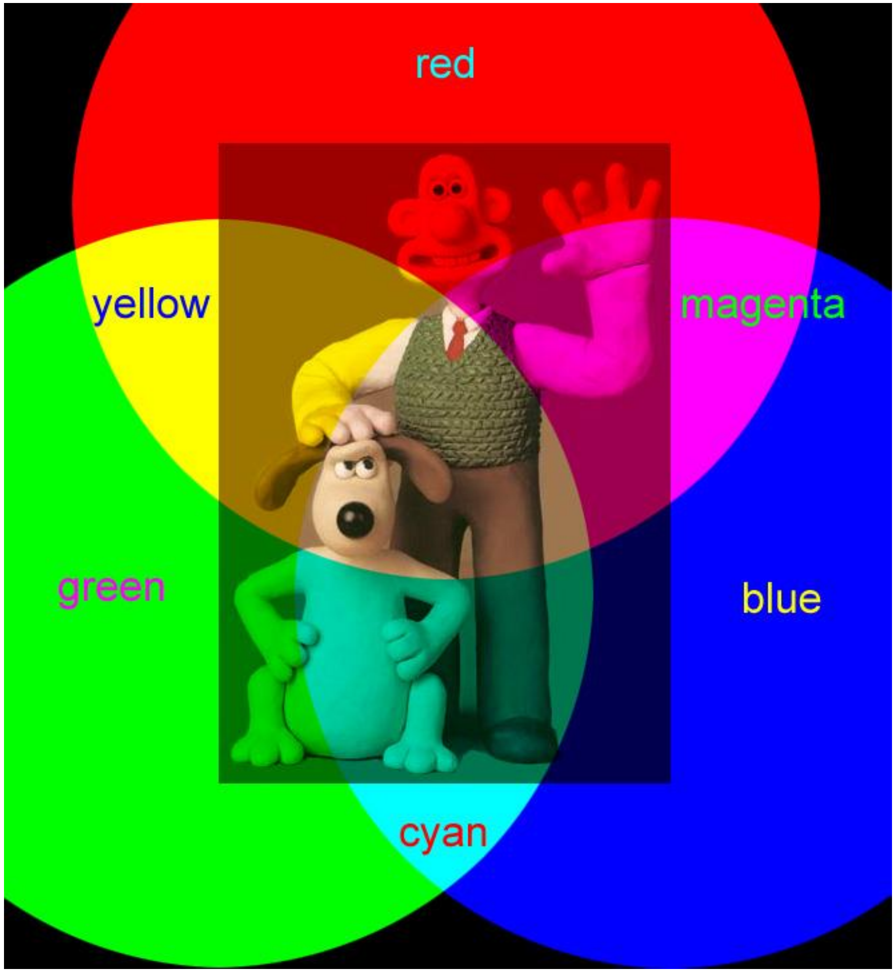
Richard Alan Peters II

Department of Electrical Engineering and Computer Science

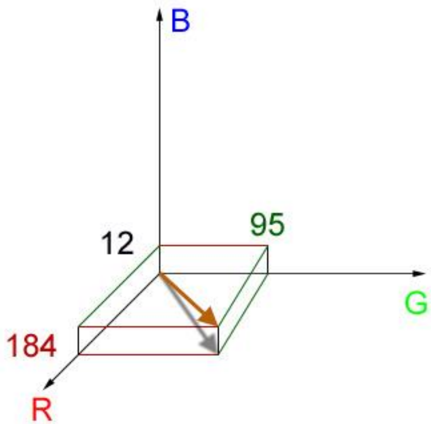
Fall Semester 2016

Color Images

- Are constructed from three overlaid intensity maps.
- Each map represents the intensity of a different "primary" color.
- The actual hues of the primaries do not matter as long as they are distinct.
- The primaries are 3 vectors (or axes) that form a "basis" of the color space.



Vector-Valued Pixels

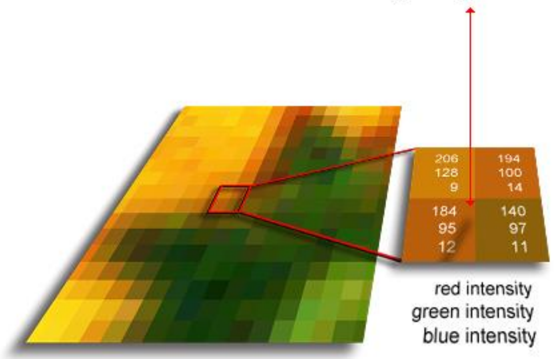


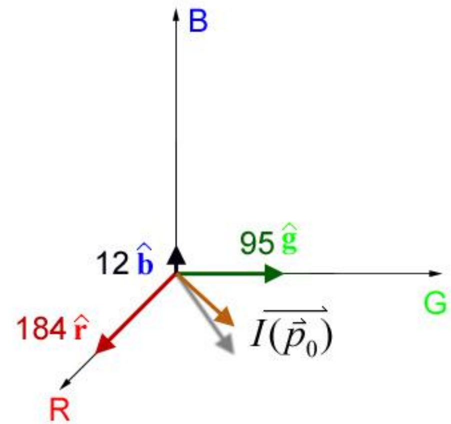
$$I(\vec{p}_0) = \begin{bmatrix} r_0 \\ g_0 \\ b_0 \end{bmatrix} = \begin{bmatrix} 184 \\ 95 \\ 12 \end{bmatrix}$$

Color Coordinates

Pixel Values

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





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

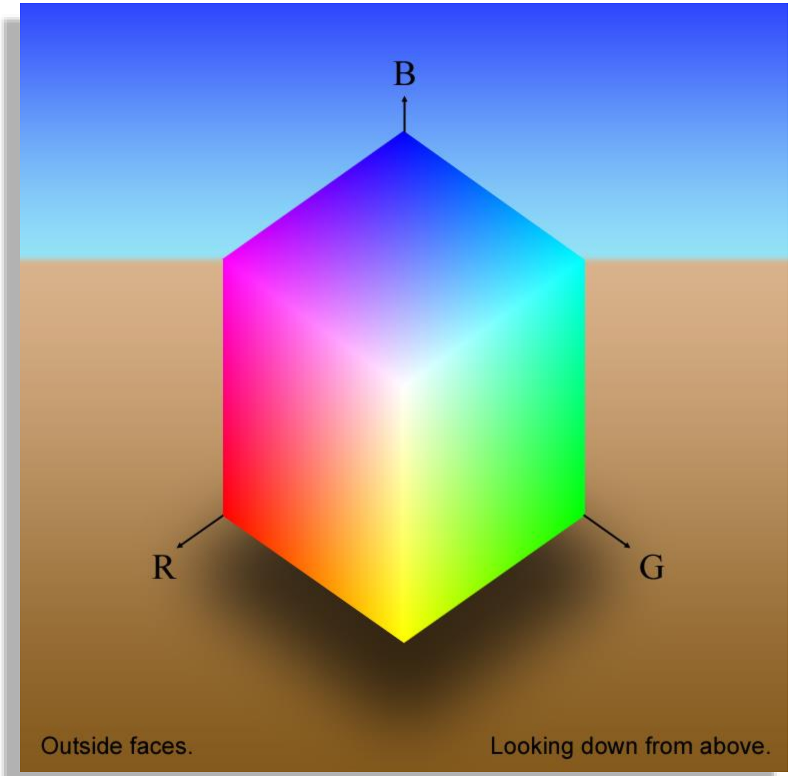
Color Vectors

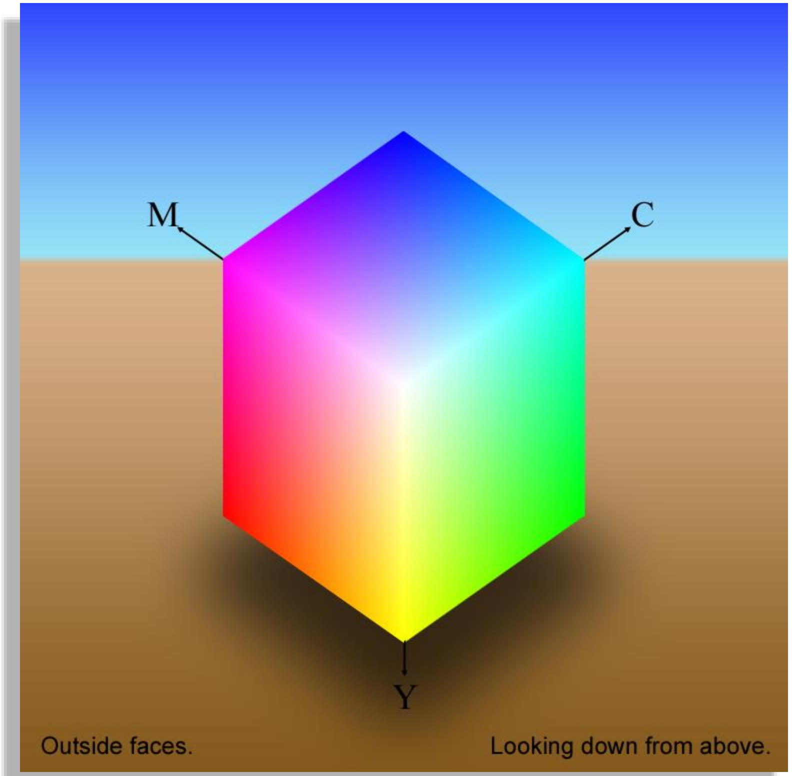
Each color corresponds to a point in a 3D vector space

RGB Color Space for standard digital images

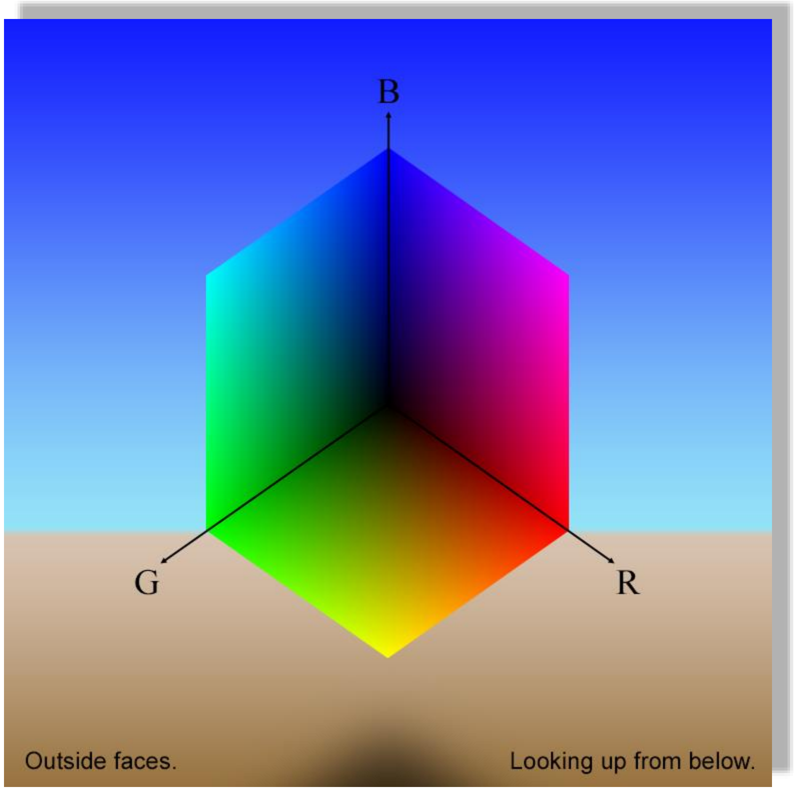
- primary image colors red, green, and blue
  - correspond to R,G, and B axes in color space.
- 8-bits of intensity resolution per color
  - correspond to integers 0 through 255 on axes.
- no negative values
  - color "space" is a cube in the first octant of 3-space.
- color space is discrete
  - 256^3 possible colors = 16,777,216 elements in cube.

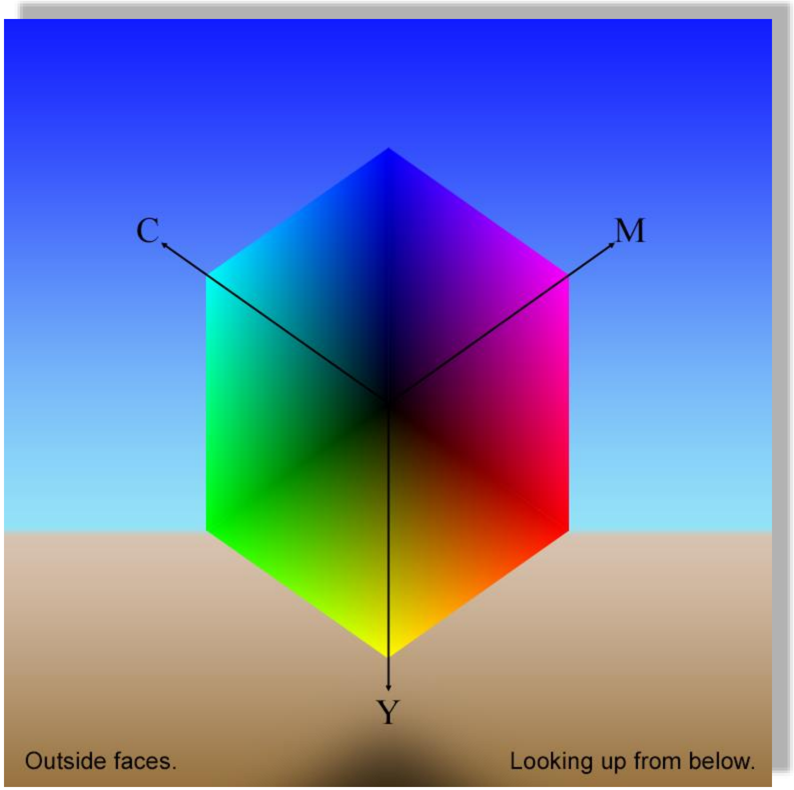
Color Cube: Faces (White-Point View)



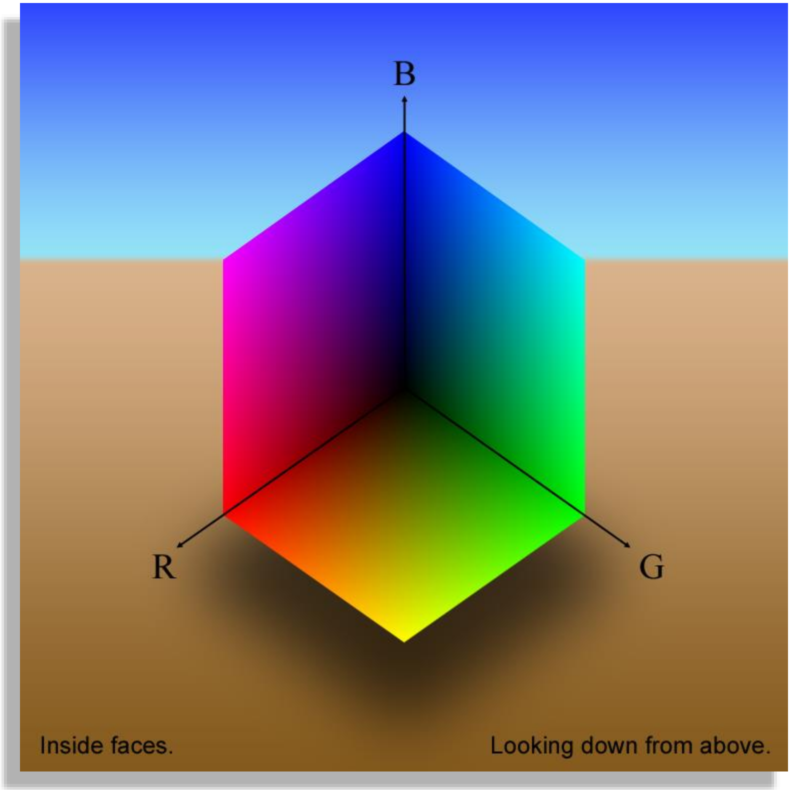


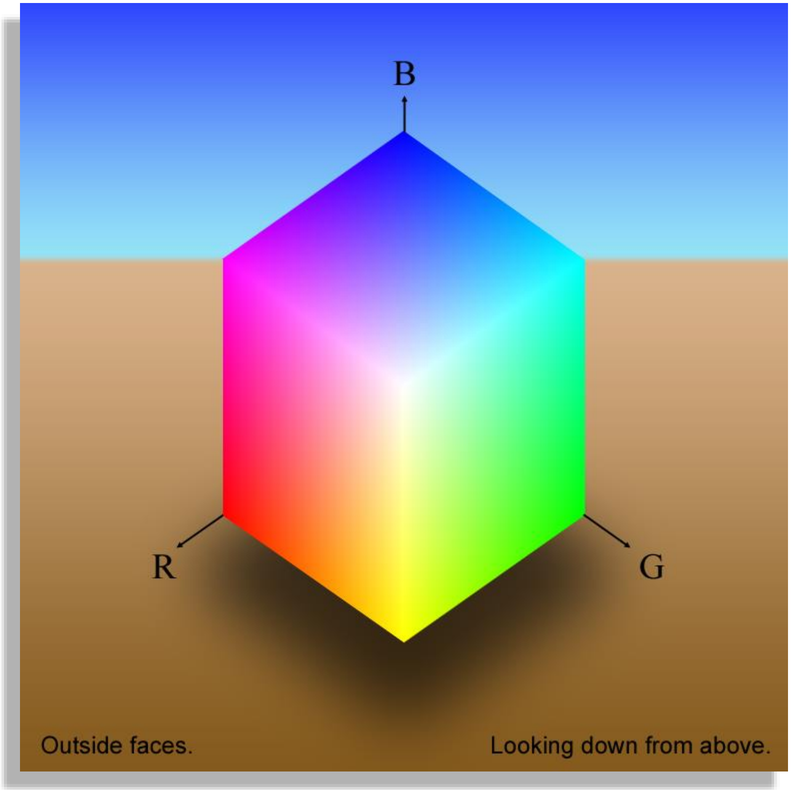
Color Cube: Faces (Black-Point View)



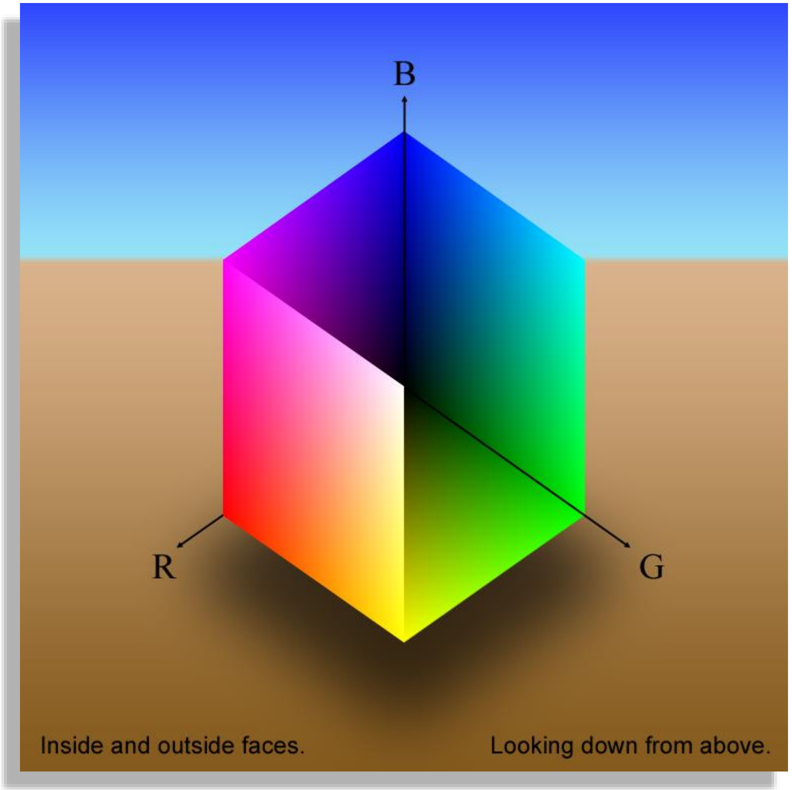


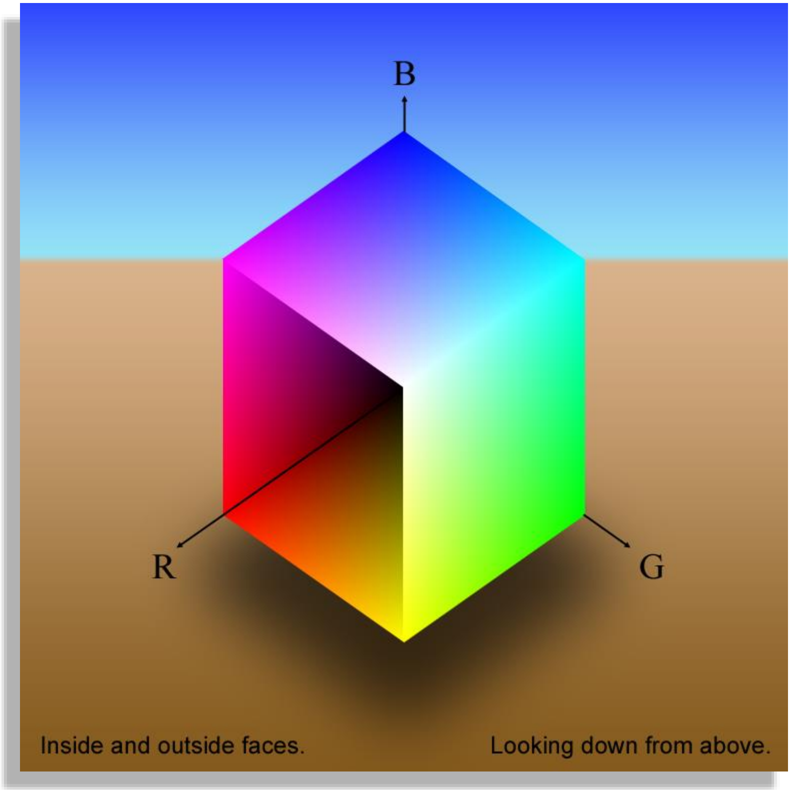
Color Cube: Faces (inner and outer)



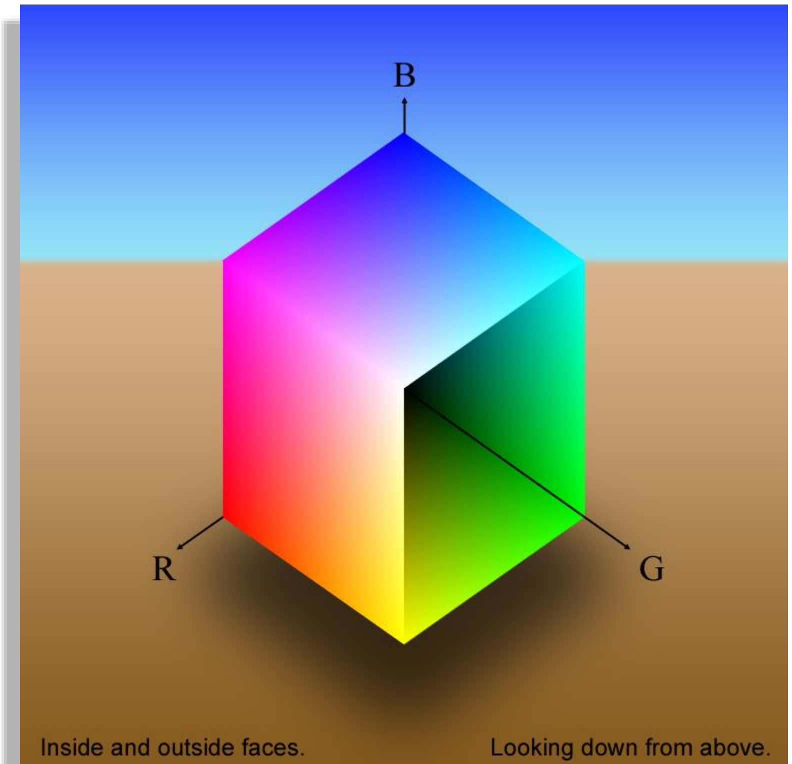


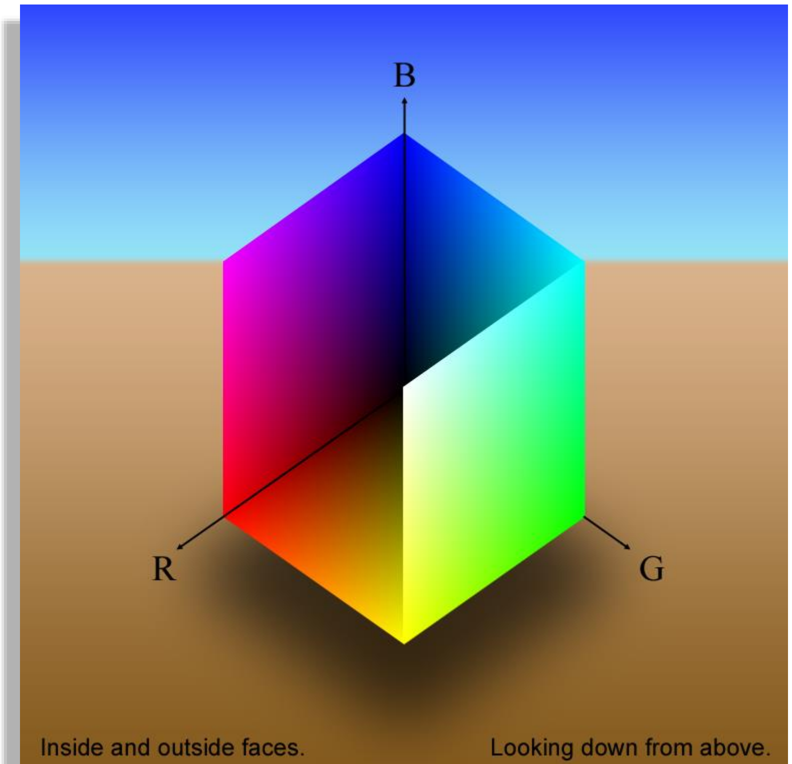
Color Cube: Faces (inner and outer)



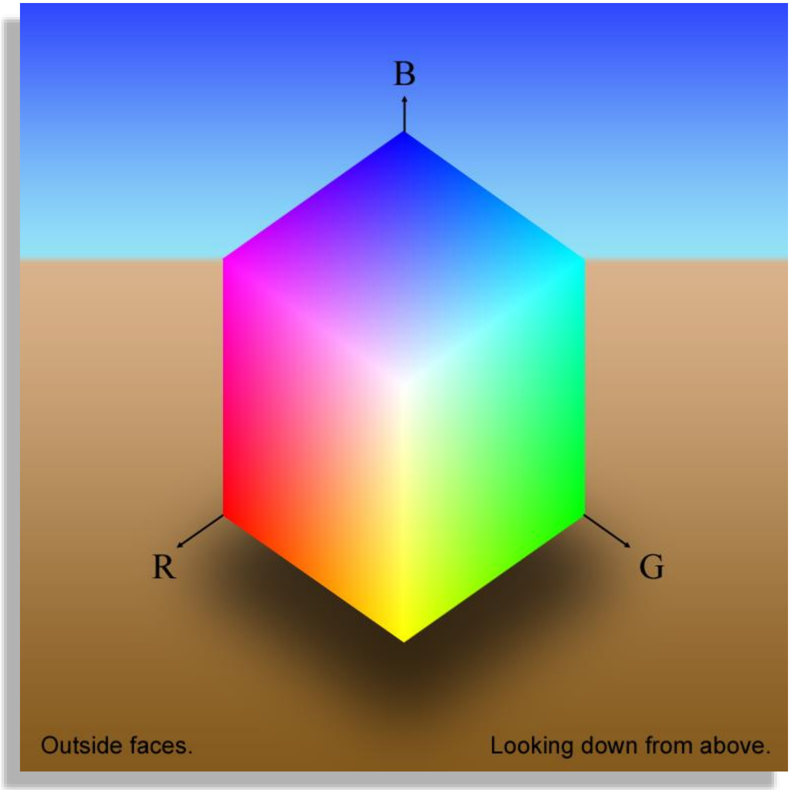


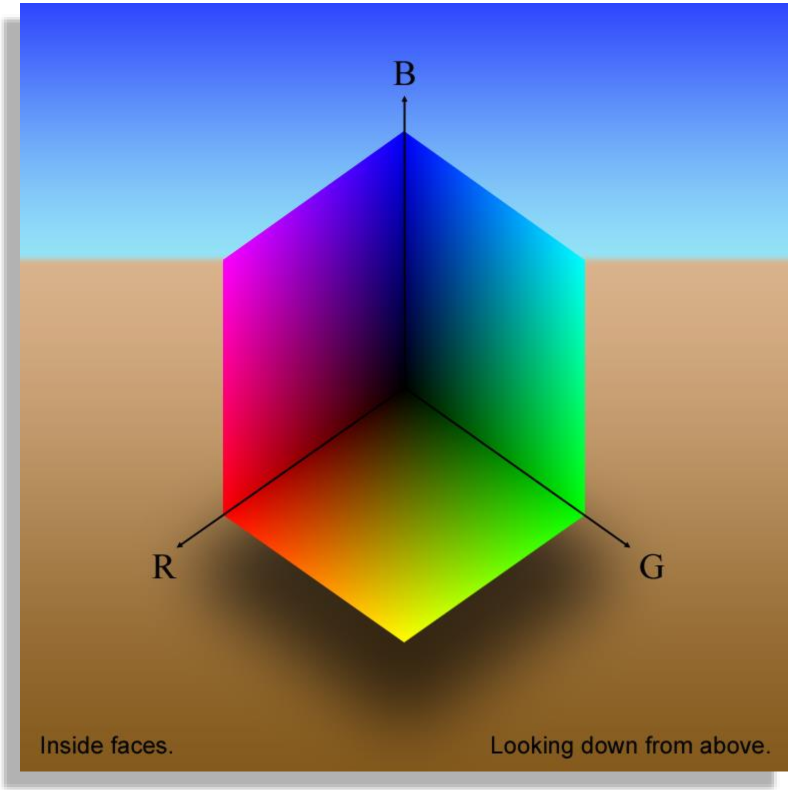
Color Cube: Faces (inner and outer)



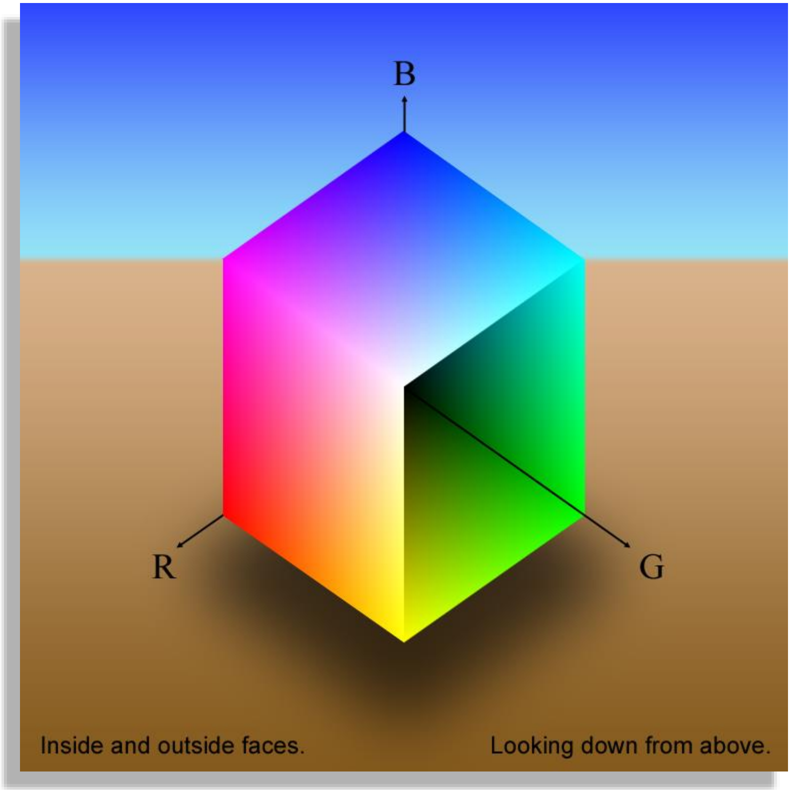


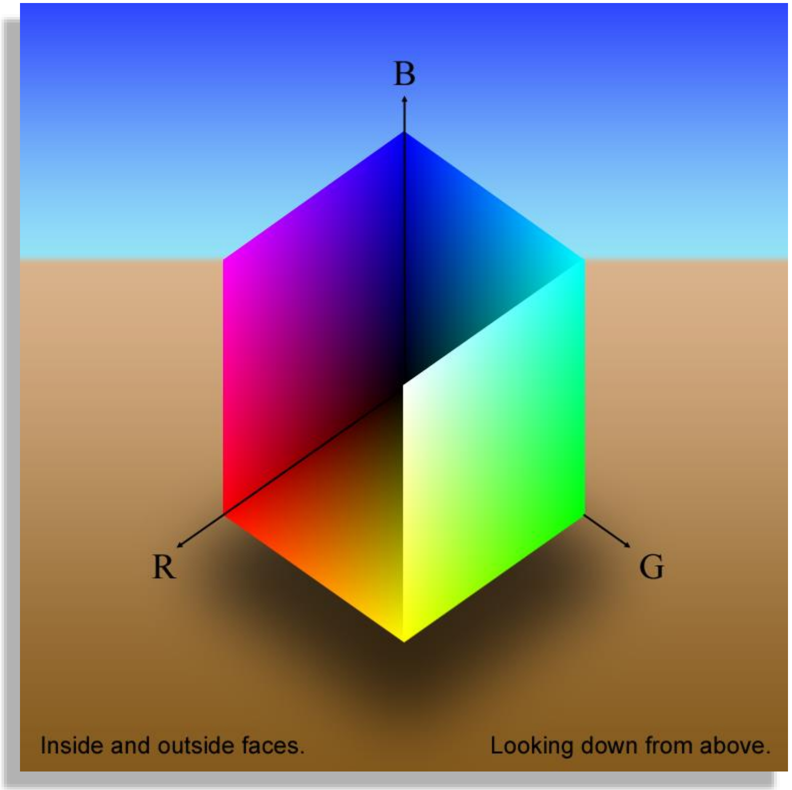
Color Cube: Faces (inner and outer)



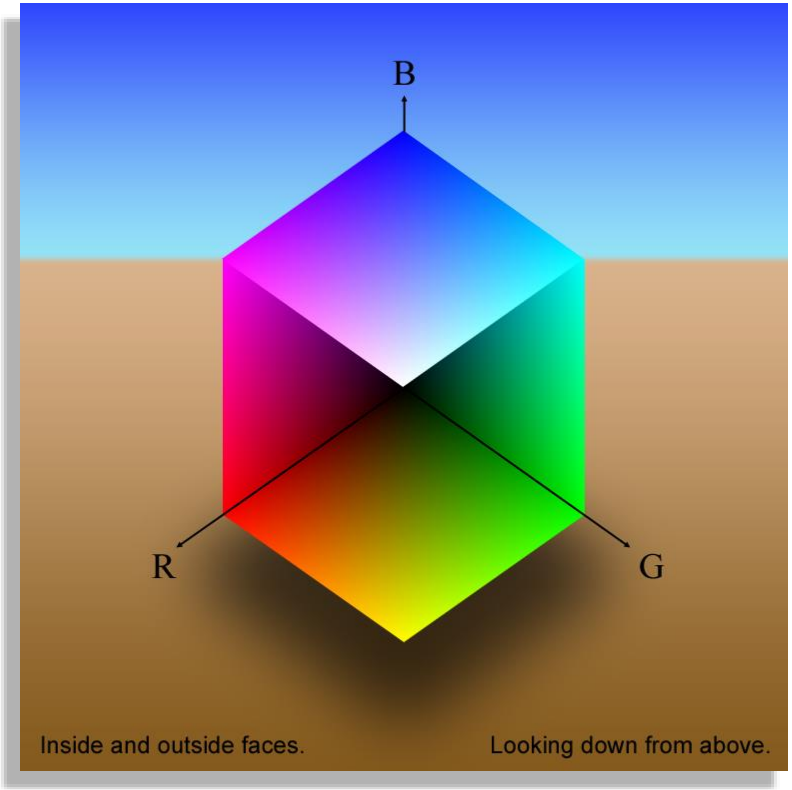


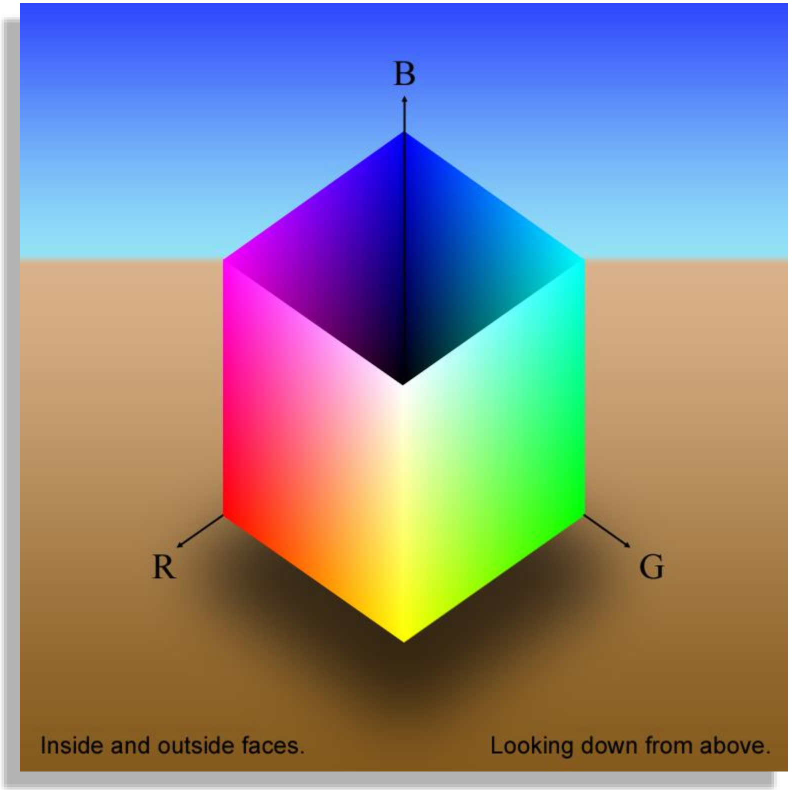
Color Cube: Faces (inner and outer)



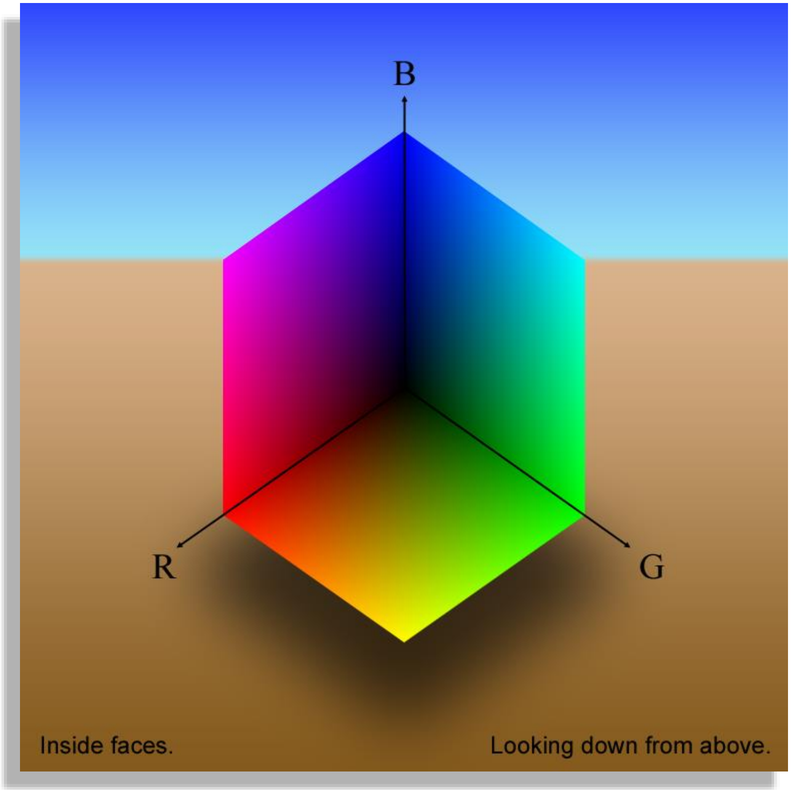


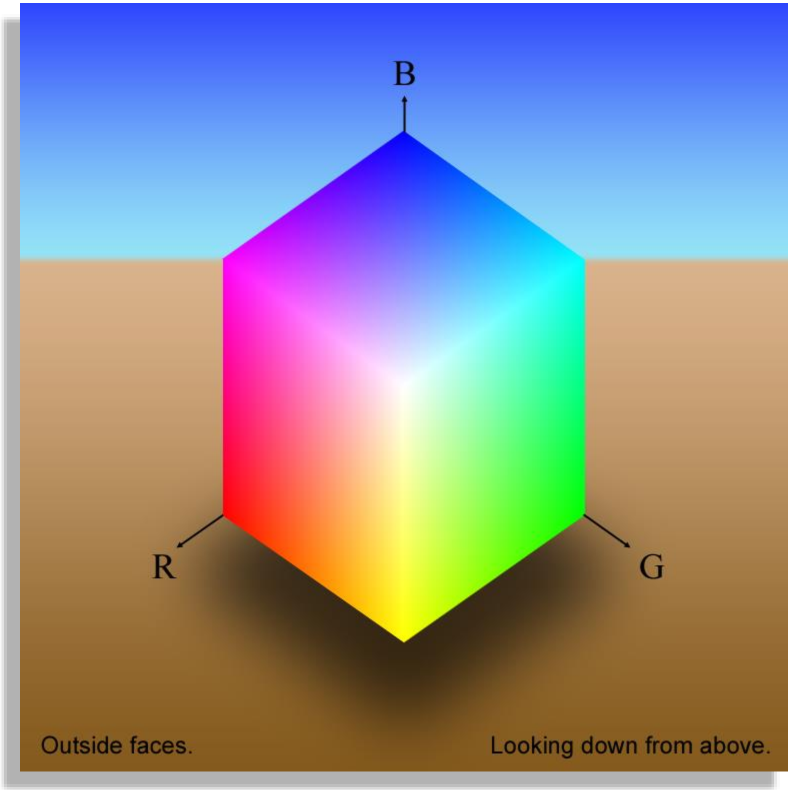
Color Cube: Faces (inner and outer)



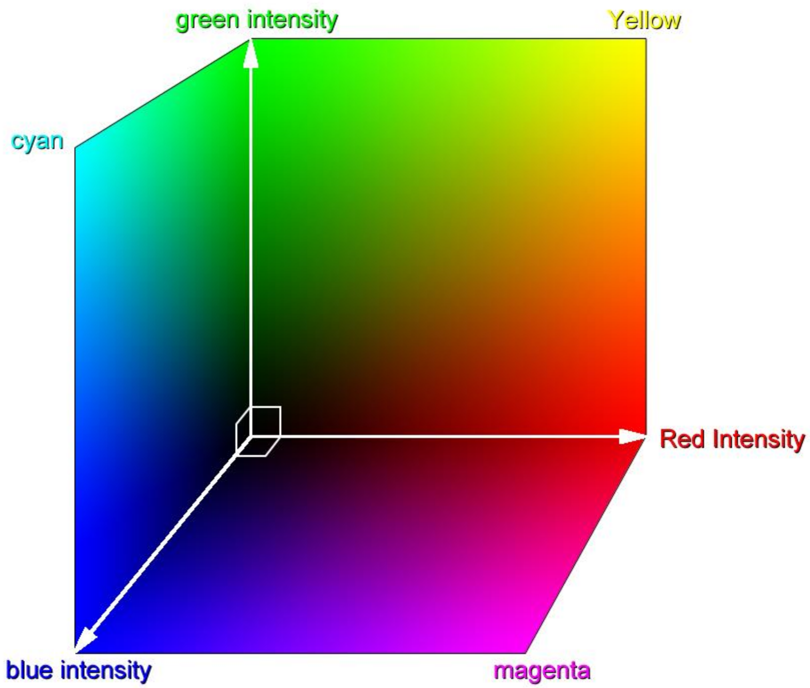


Color Cube: Faces (inner and outer)

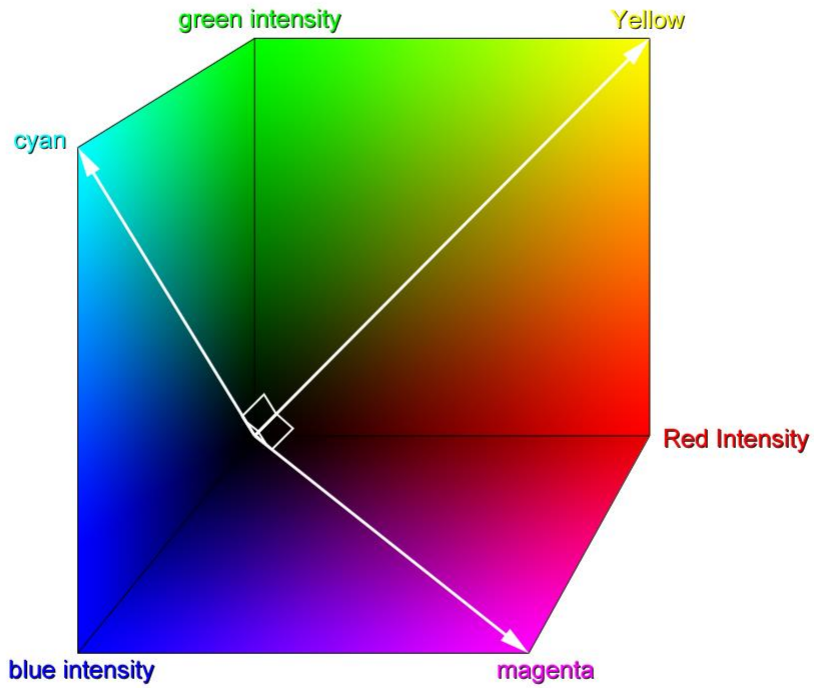




Different Axis Sets in Color Space

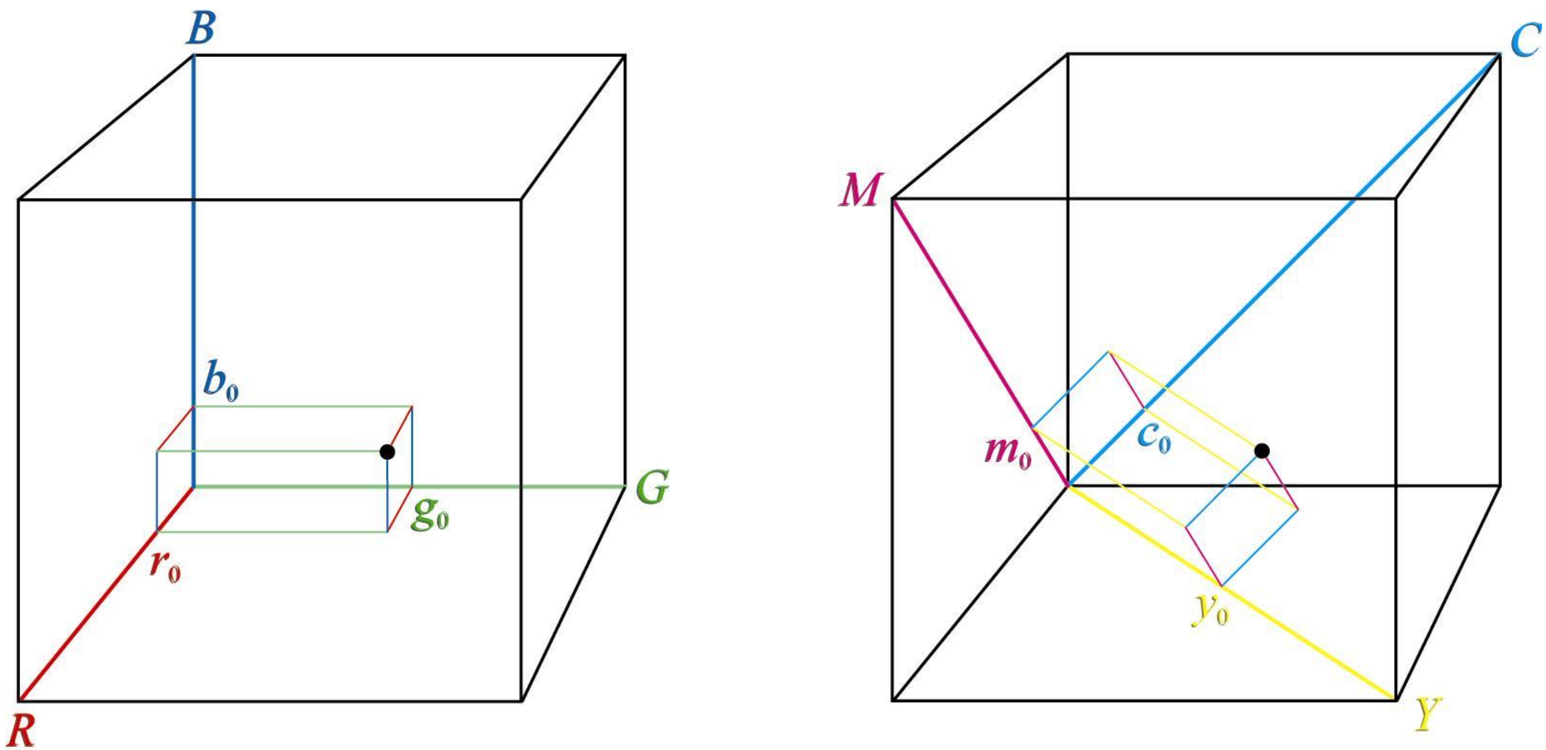


RGB axes



CMY axes

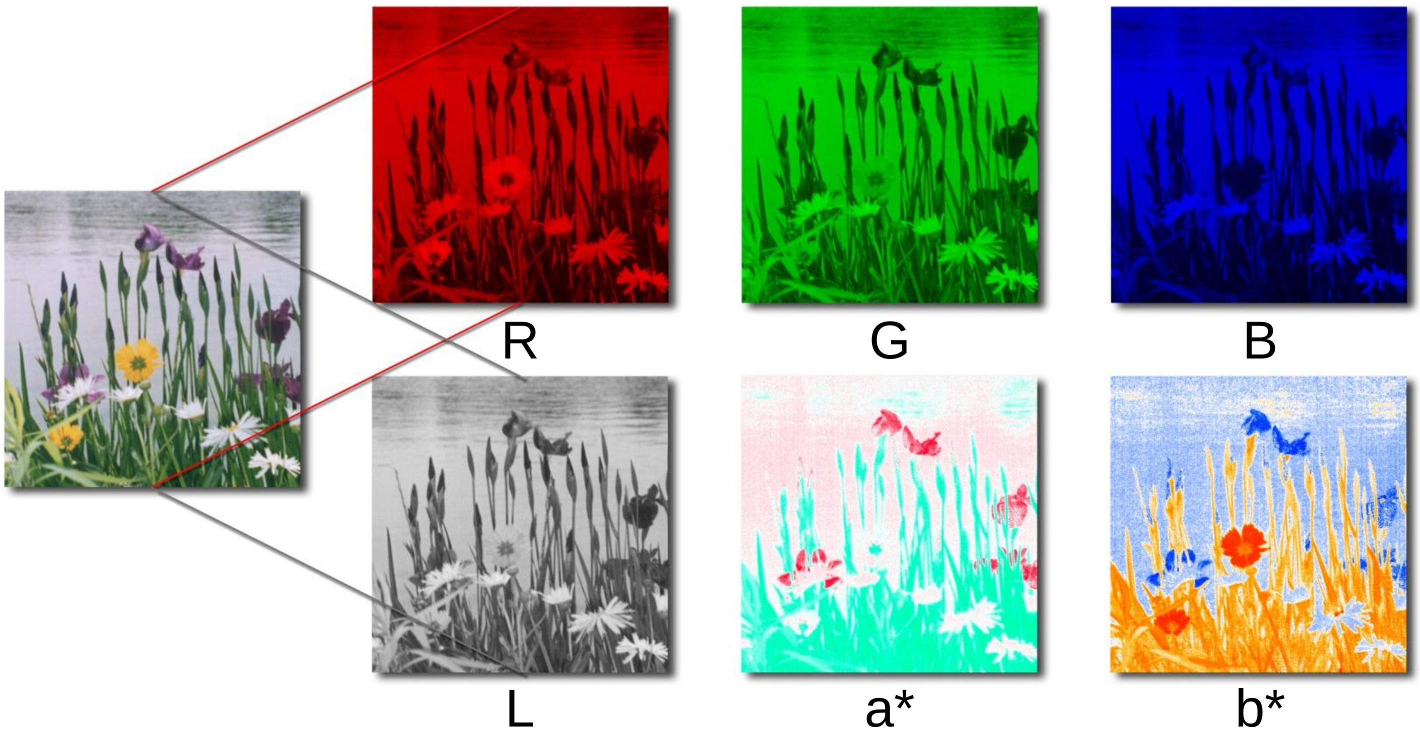
Color With Respect To Different Axes



The same color has different RGB and CMY coordinates.

Color Images

are represented by three bands (not uniquely) *e.g.*, R, G, & B or L, a*, & b*.

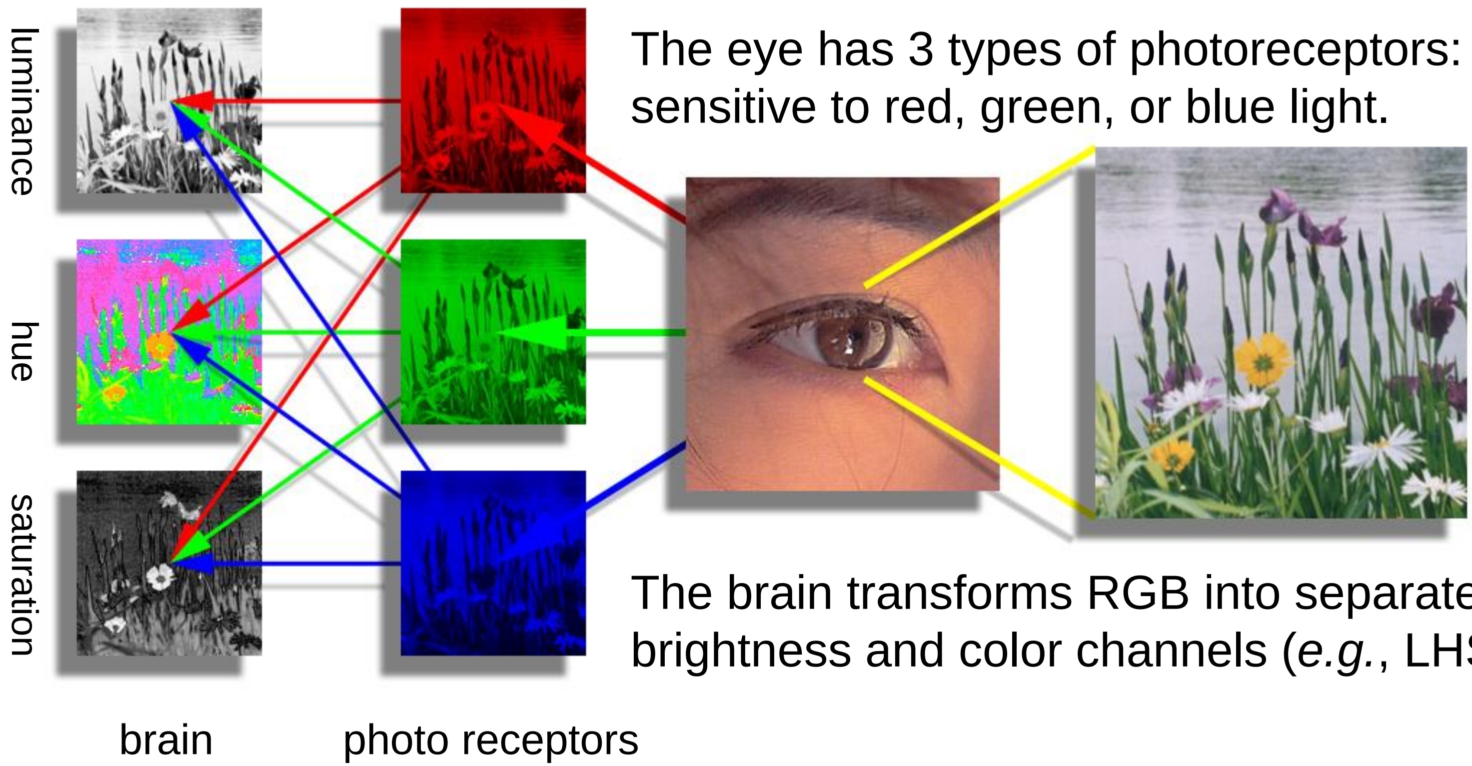


Red
Green
Blue

Luminance
a*-chroma
b*-chroma

RGB to LHS: A Perceptual Transformation

The eye has 3 types of photoreceptors: sensitive to red, green, or blue light.



The brain transforms RGB into separate brightness and color channels (*e.g.*, LHS).

Color Representations: Color Spaces

RGB is the most familiar color representation because our digital imaging devices use them to encode and to display images. Because they have sensors that are sensitive to red, green, and blue light, cameras transduce an image projected onto its sensor array into a matrix of 3-vectors.

The vectors may be transformed into other representations for processing or analysis. Here we consider one such representation, the HSV colorspace. It is a cylindrical coordinate representation of colors that separates the brightness (value) from the chrominance (hue and saturation).

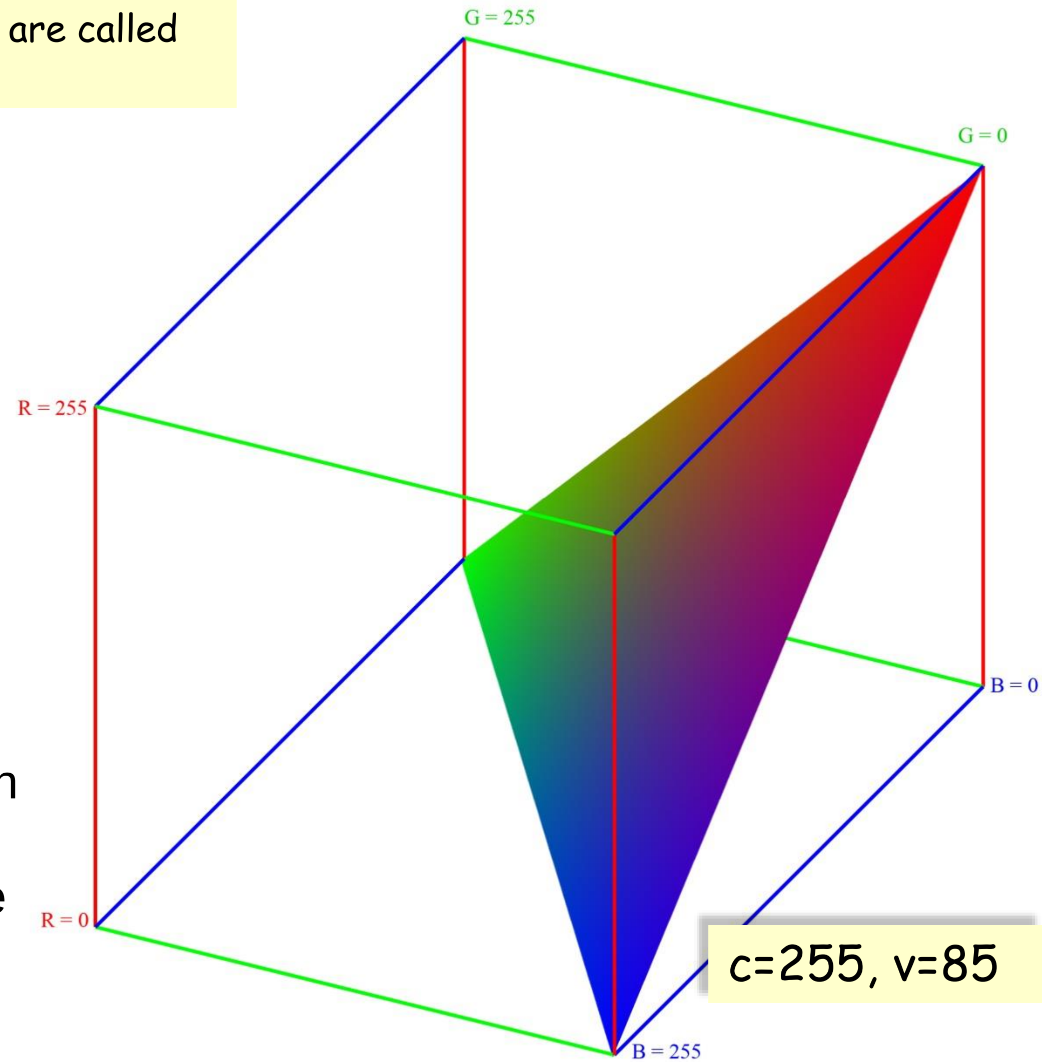
HSV is defined with respect to planes that cut the cube perpendicularly to the gray line – $c\cdot[1\ 1\ 1]^T$ where $c \in \{0,\ldots,255\}$. These are called equivalue planes.

Equivalue Color Triangle

A plane through the colors

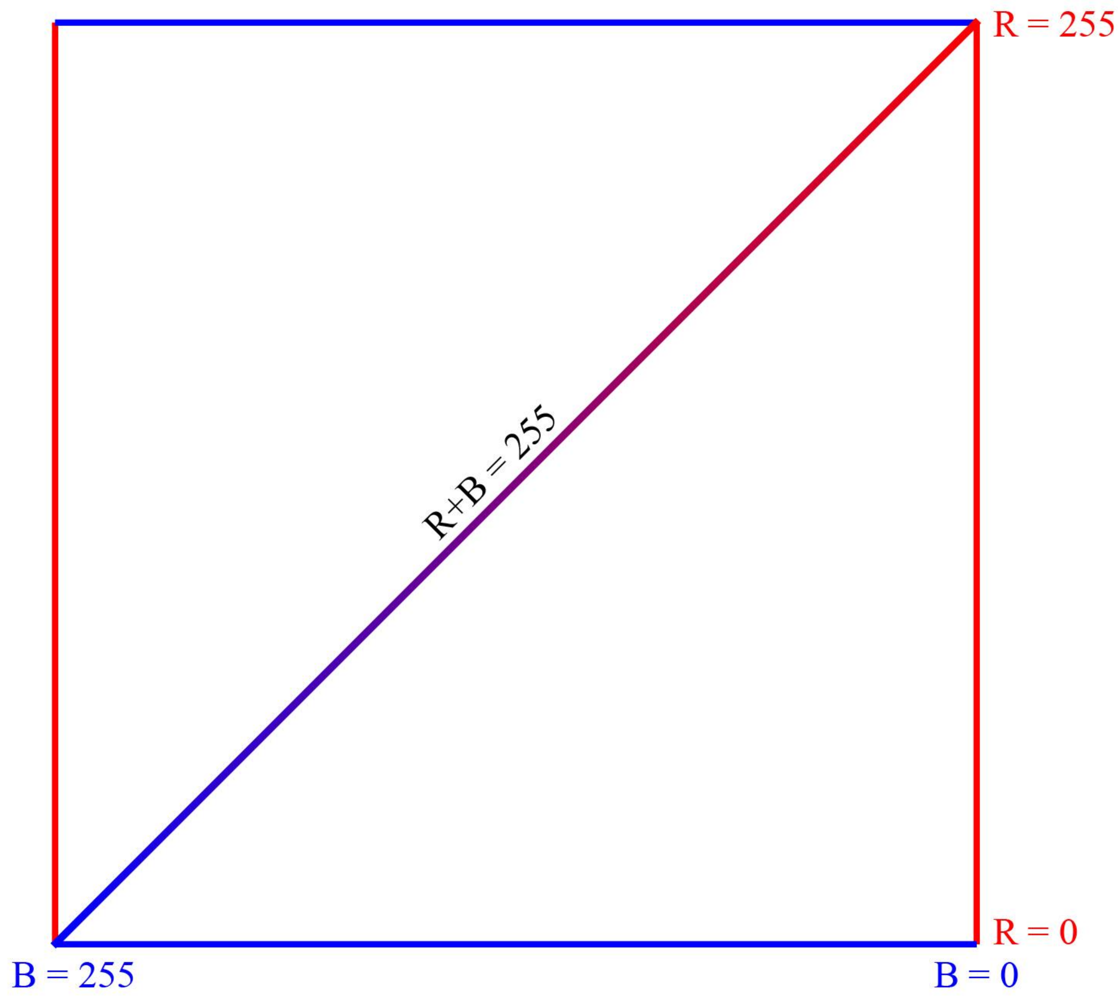
$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} c \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ c \\ 0 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix},$$

forms a **triangle** inside the color cube if $c \leq 255$ or $c \geq 510$, or a hexagon if $255 < c < 510$. Every color on the planar surface is such that $r + g + b = c.$ Therefore its value is $c/3$. It is on this equivalue plane that hue and saturation are computed.



Equivalue Color Triangle

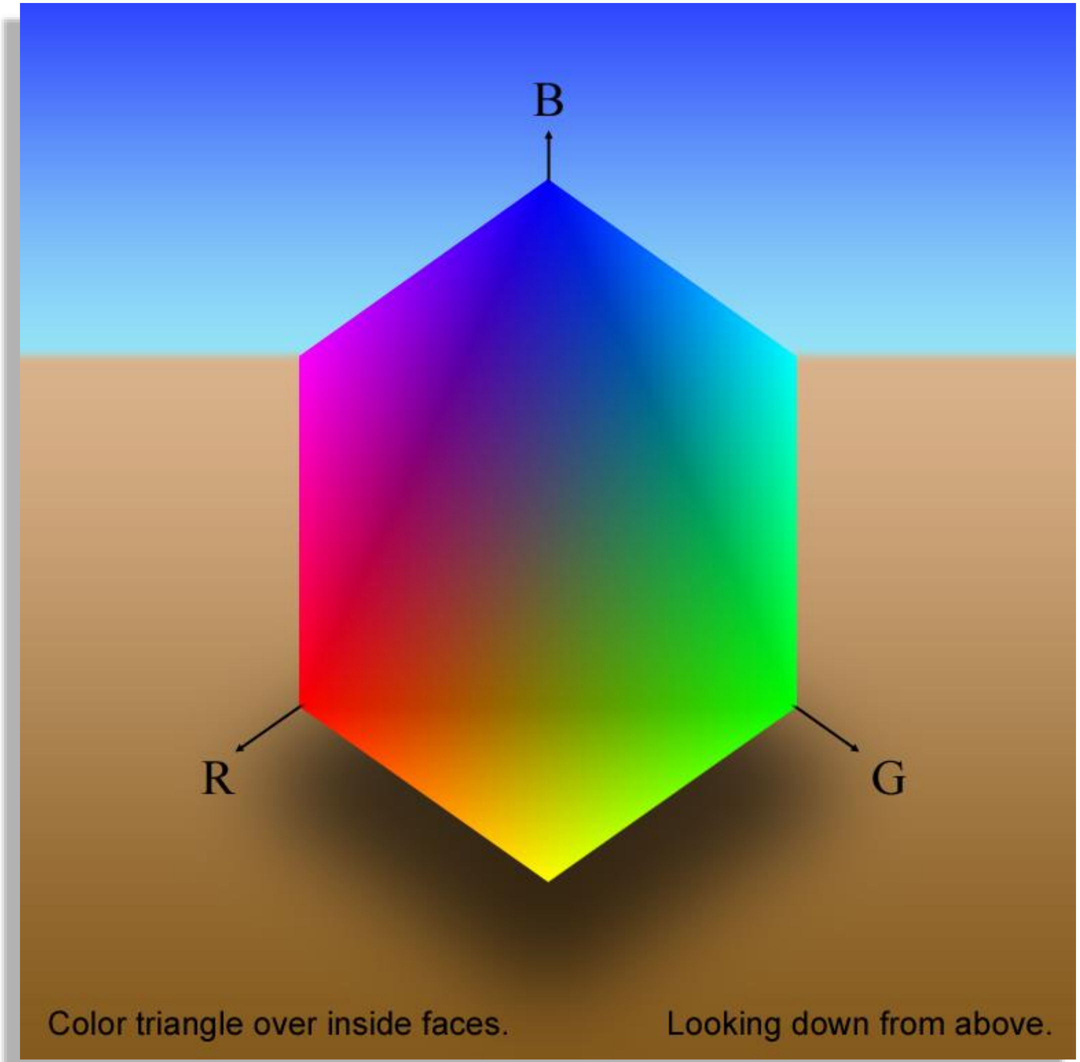
c=255, v=85



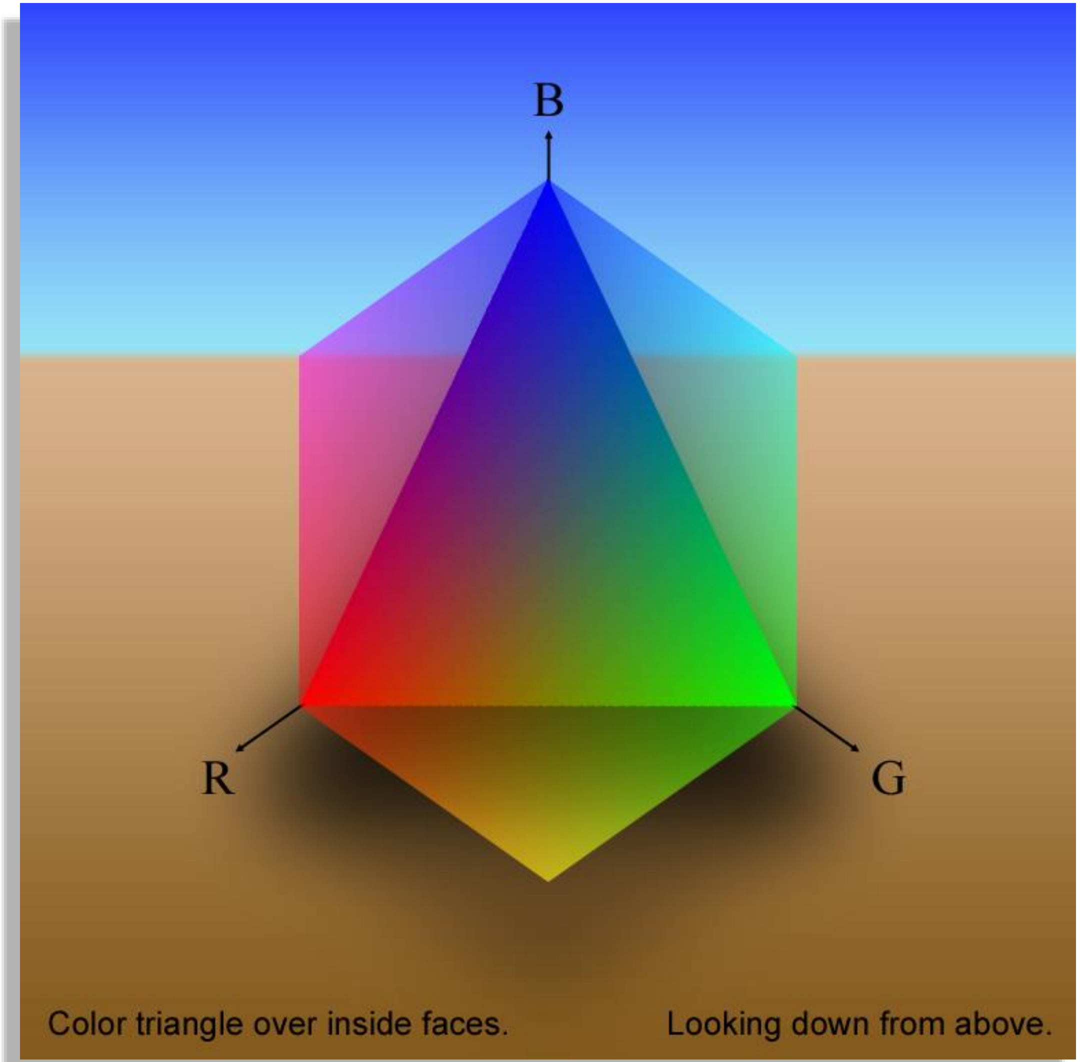
On the $g = 0$ face of the cube the triangle traces the line, $r + b = 255$.

Color Cube: Equivalue Triangle

Notice the dark outline of the triangle. It is not there – not a real feature of the color surface. The planar discontinuities in 3D at the edges of the triangle project to abrupt changes in the color gradient in the 2D image. The change in gradient is a second order effect that is exaggerated by the color-agonist, center-surround edge detectors on the retina. It is an optical illusion.

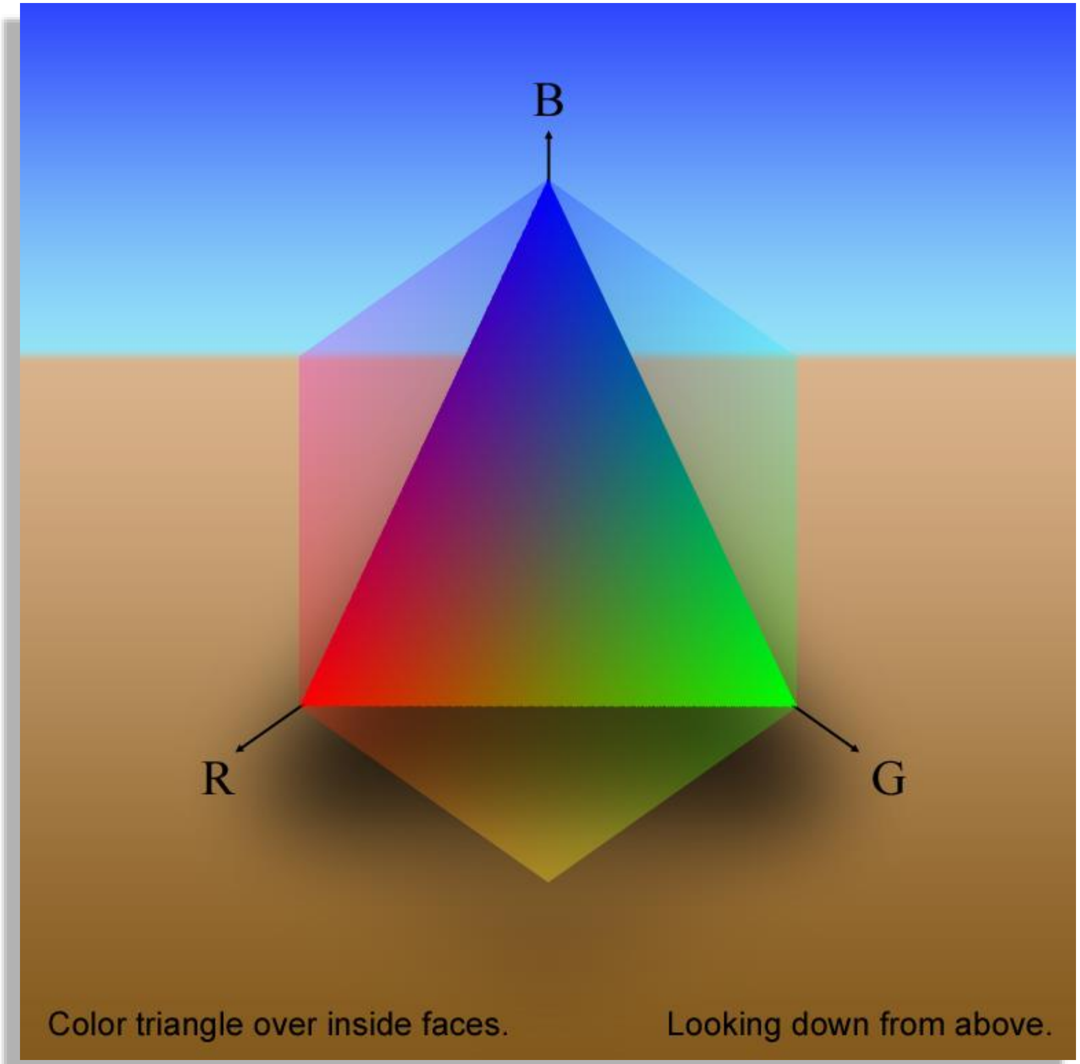


Color Cube: Equivalue Triangle

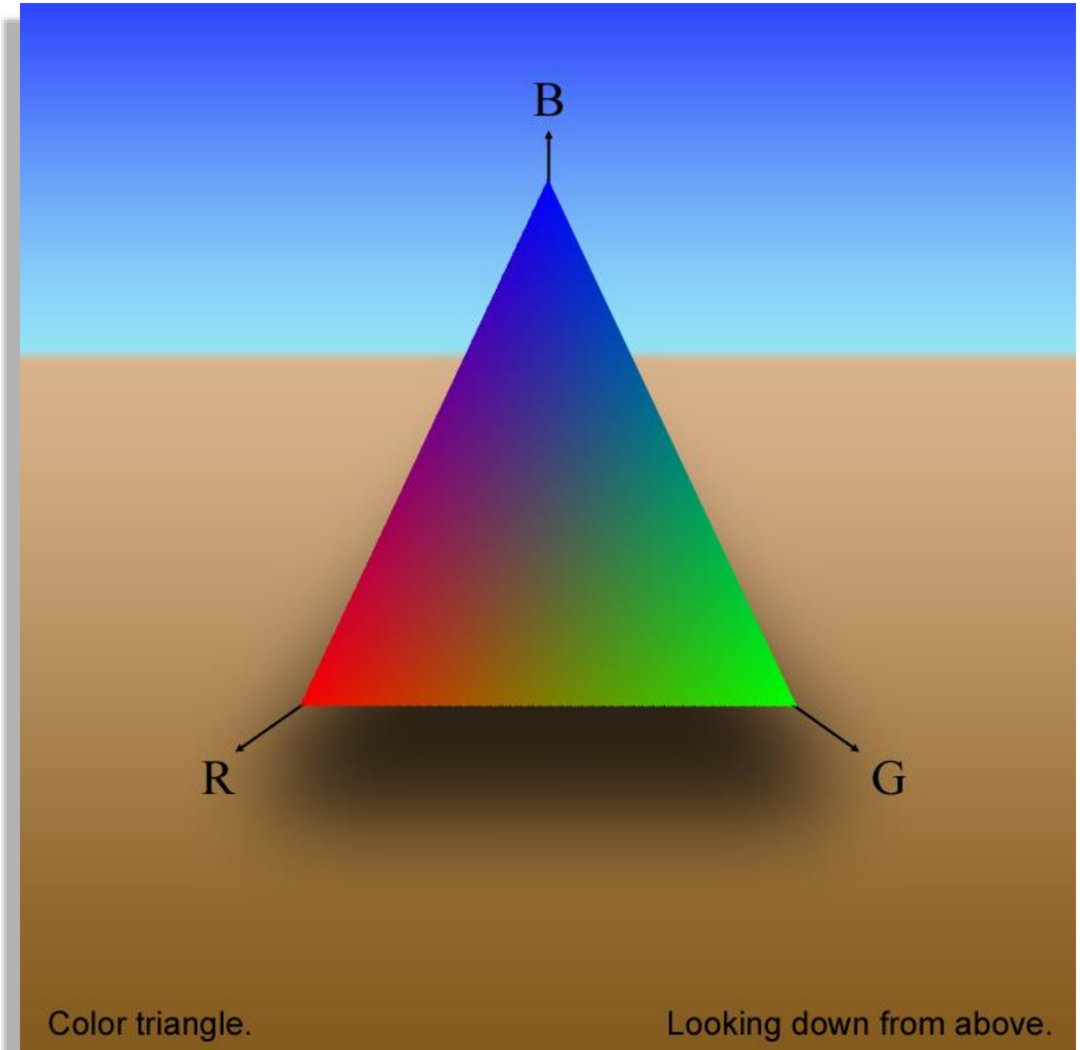


As the cube is faded out over this and the next 2 slides, the discontinuity diminishes and dark outline disappears.

Color Cube: Equivalue Triangle



Color Cube: Equivalue Triangle



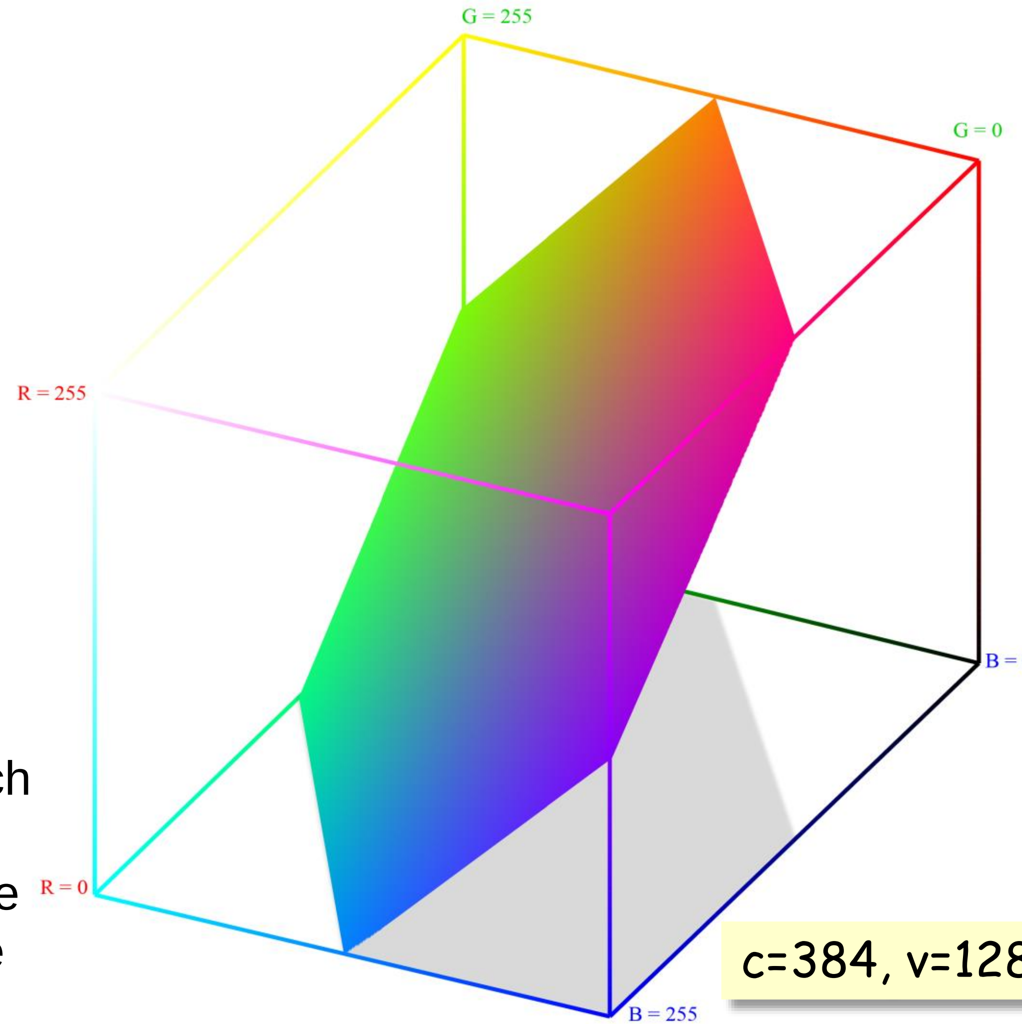
Every point on the equivalue plane has the same value. (“well, duh!”) There is one plane for each value (r+g+b)/3. The hue and saturation for all colors with the same value are calculated in the plane.

Equivalue Color Hexagon

A plane through the colors

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

forms a triangle inside the color cube if $c \leq 255$ or $c \geq 510$, or a **hexagon** if $255 < c < 510$. Every color on the planar surface is such that $r + g + b = c$. Therefore its value is $c/3$. It is on this equivalue plane that hue and saturation are computed.



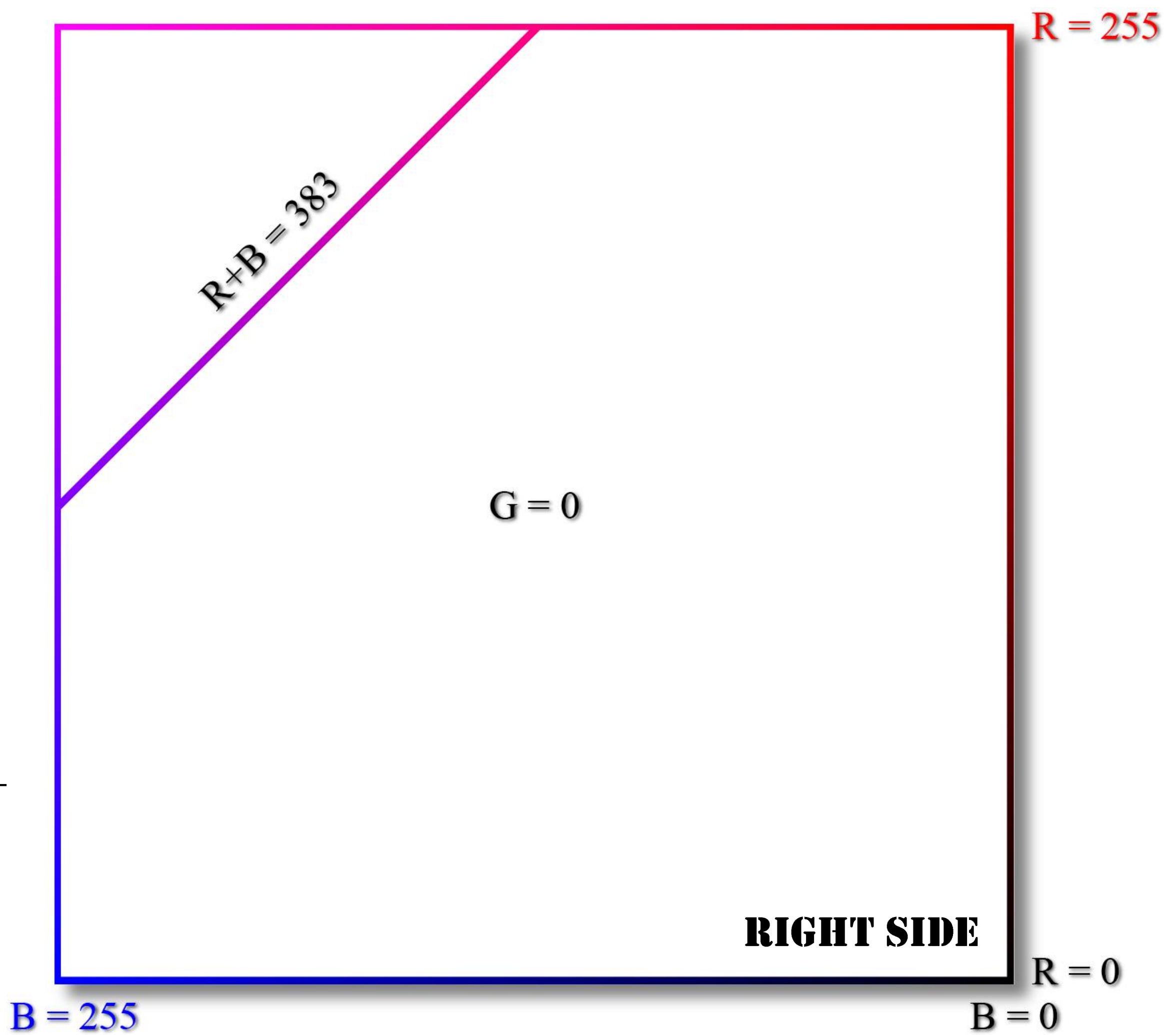
c=384, v=128

Equivalue Color Hexagon

Outside looking in.

c=383, v=128

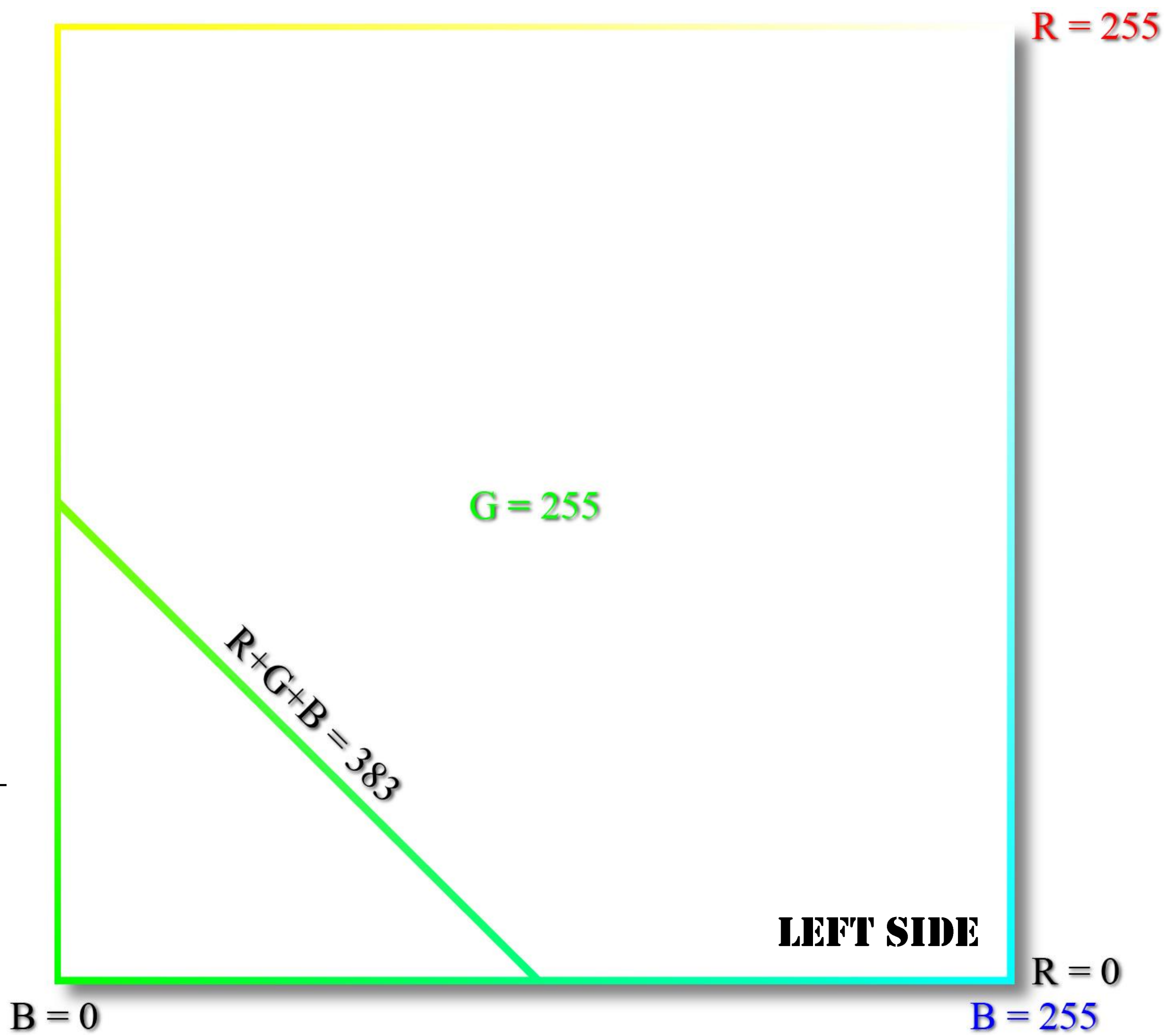
On the $g = 0$ face of the cube the hexagon traces the line, $r + b = 383$.



Equivalue Color Hexagon

Outside looking in.

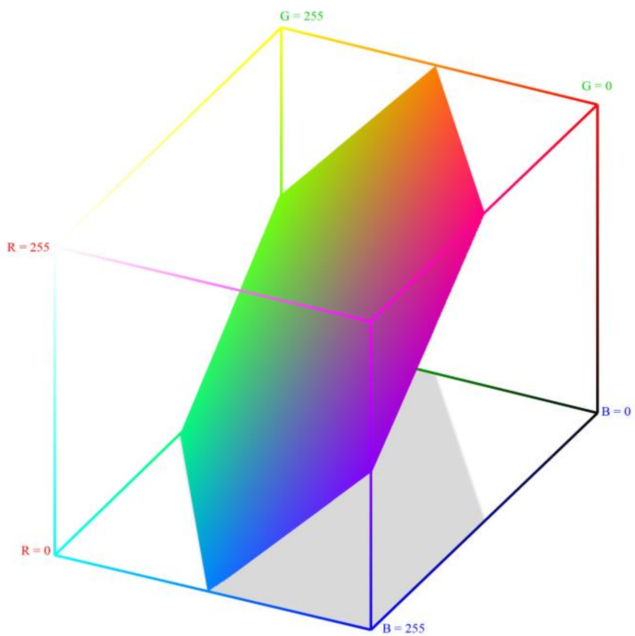
c=383, v=128

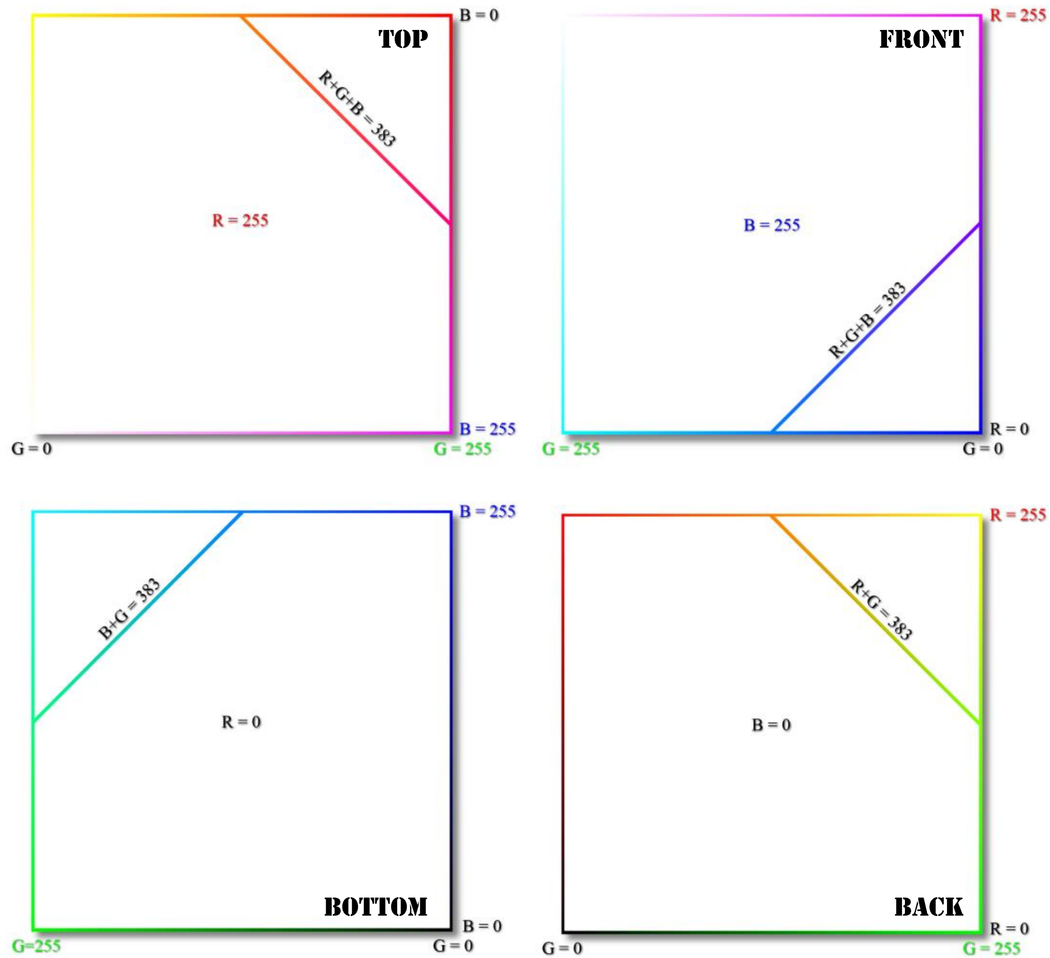


On the g = 255 face of the cube the hexagon traces the line,
$r + b + g = 383.$

Equivalue Color Hexagon

Outside looking in.

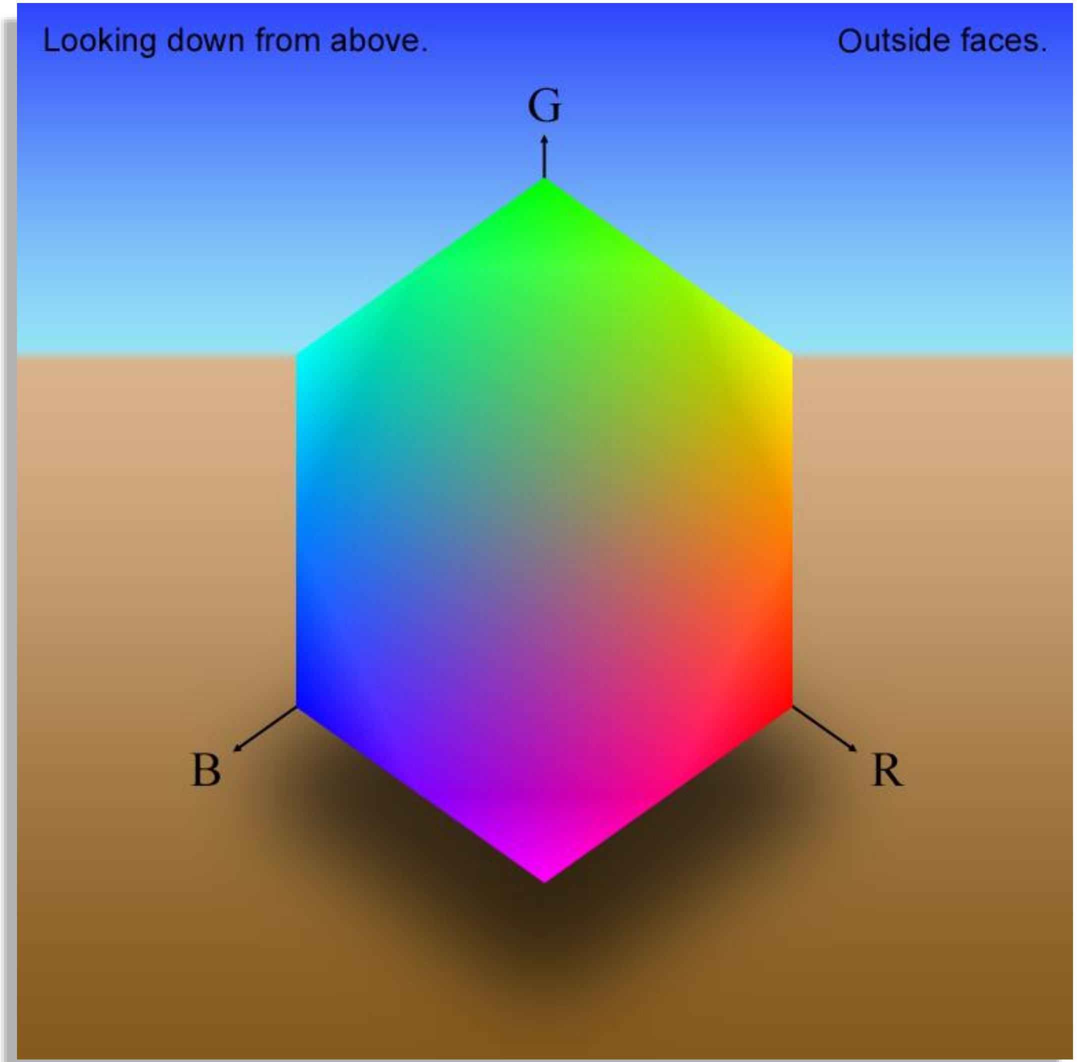




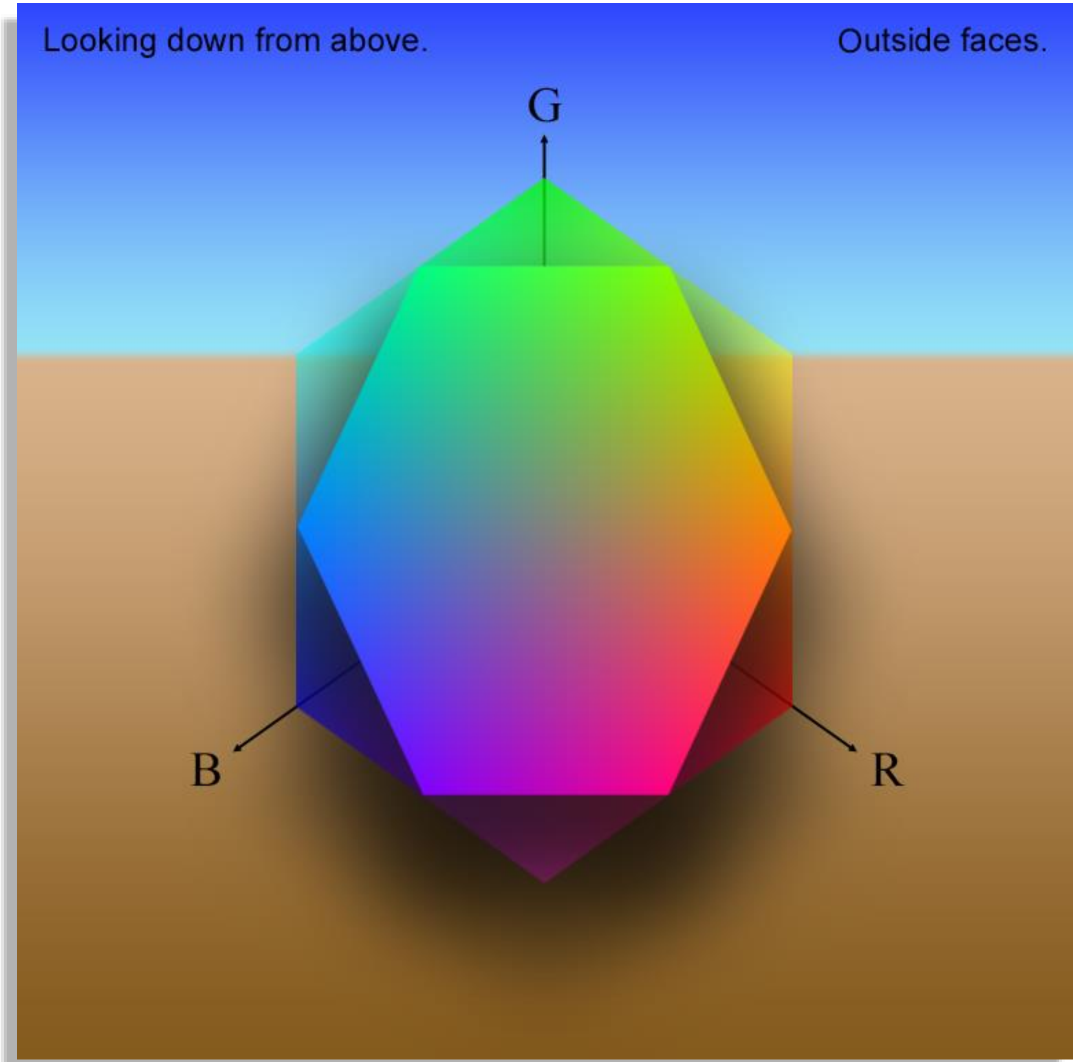
The lines on the other four faces similarly have $r + b + g = 383$.

Color Cube: Equivalue Hexagon

Notice the outline of the hexagon. It is not there – not a real feature of the color surface. The planar discontinuities in 3D at the edges of the hexagon project to abrupt changes in the color gradient in the 2D image. The change in gradient is a second order effect that is exaggerated by the color-agonist, center-surround edge detectors on the retina. It is an optical illusion.

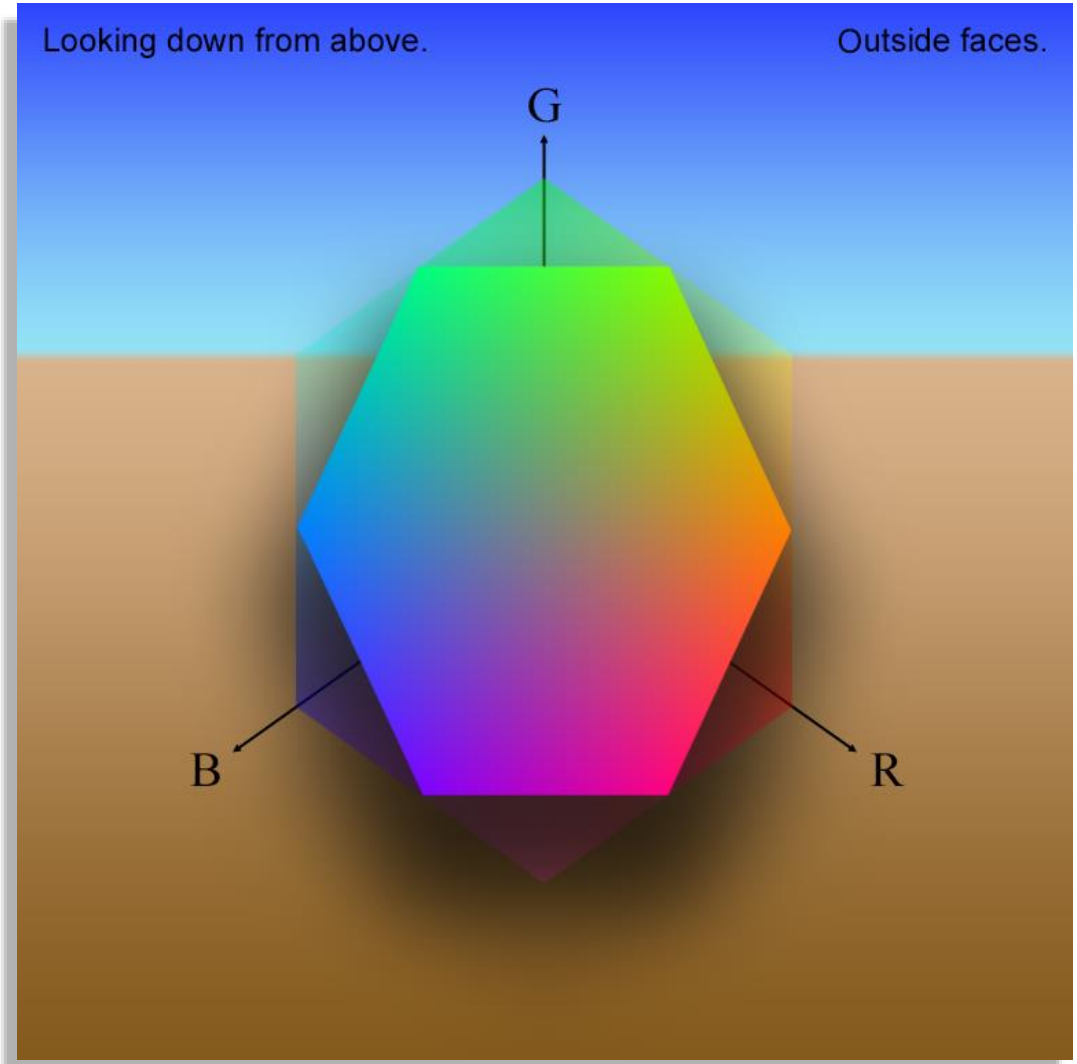


Color Cube: Equivalue Hexagon

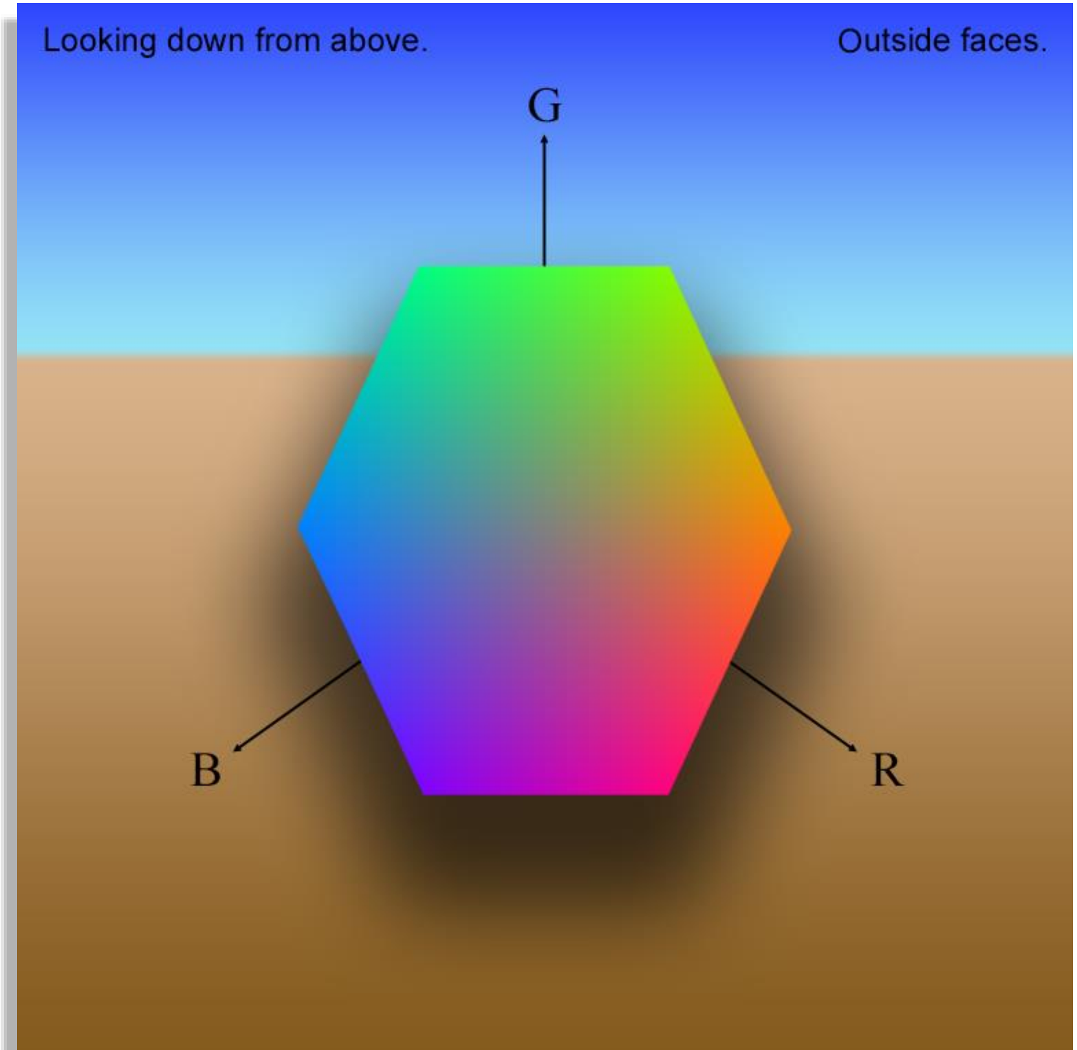


As the cube is faded out over this and the next 2 slides, the discontinuity diminishes and the virtual outline disappears.

Color Cube: Equivalue Hexagon

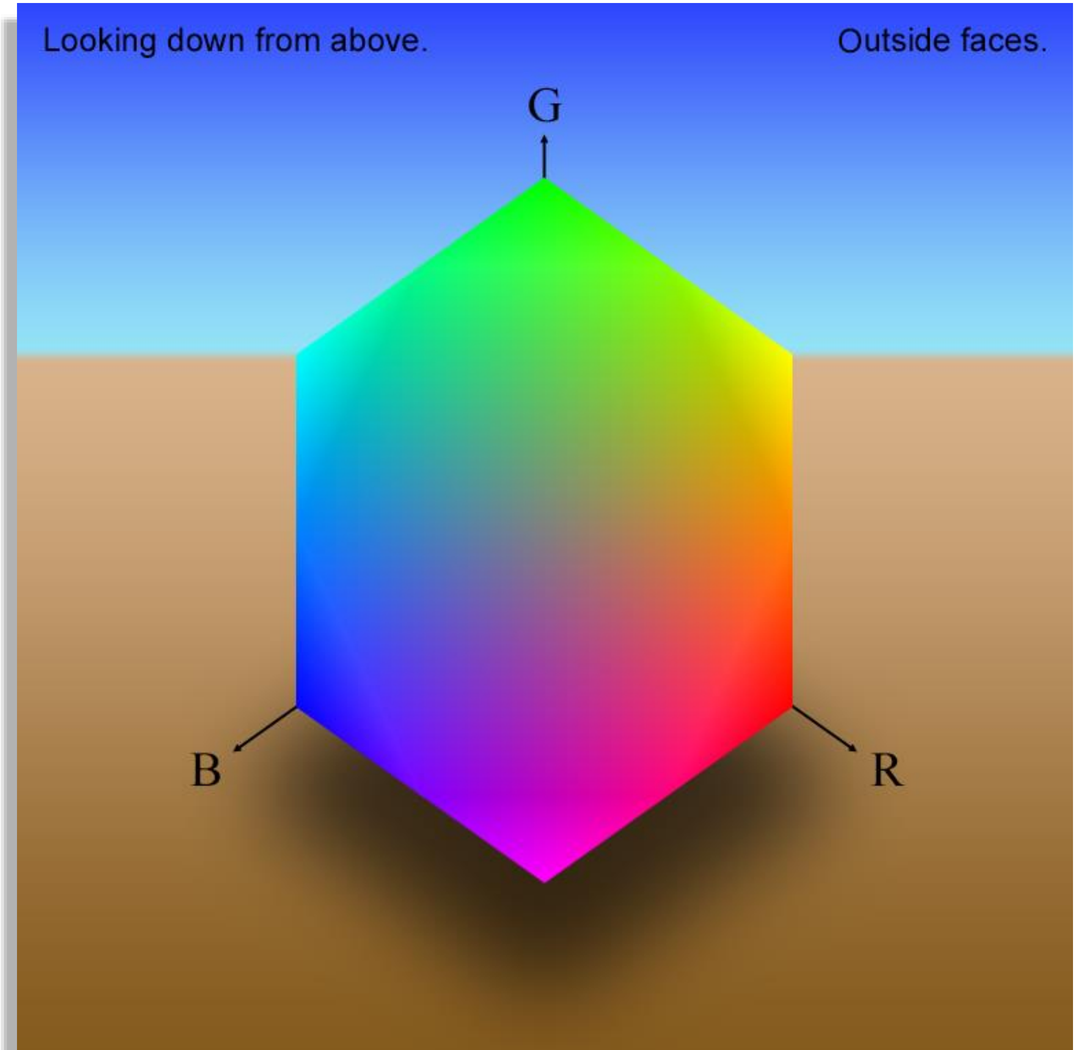


Color Cube: Equivalue Hexagon

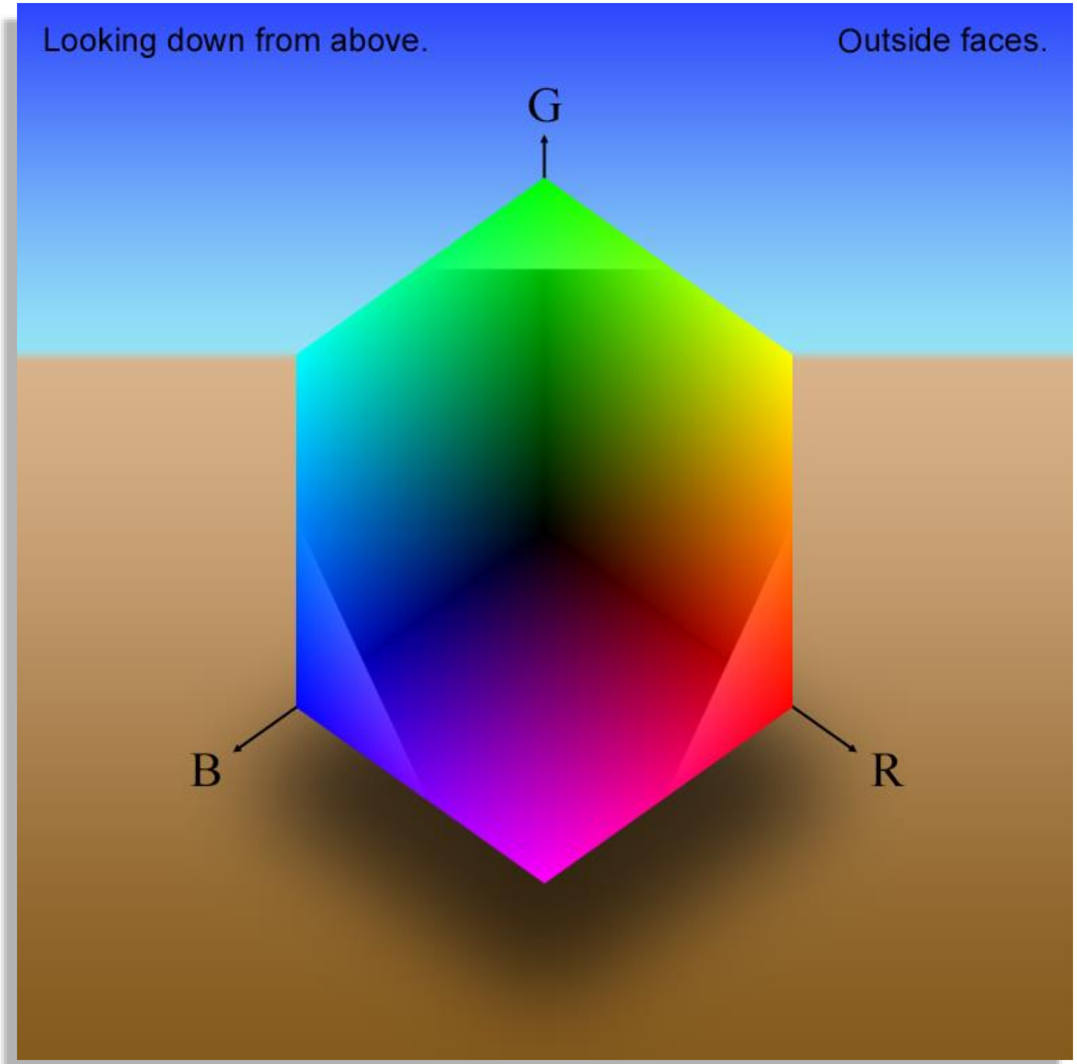


Every point on the equivalue plane has the same value. ("well, duh!") There is one plane for each value (r+g+b)/3. The hue and saturation for all colors with the same value are calculated in the plane.

Color Cube: Equivalue Hexagon



Color Cube: Equivalue Hexagon

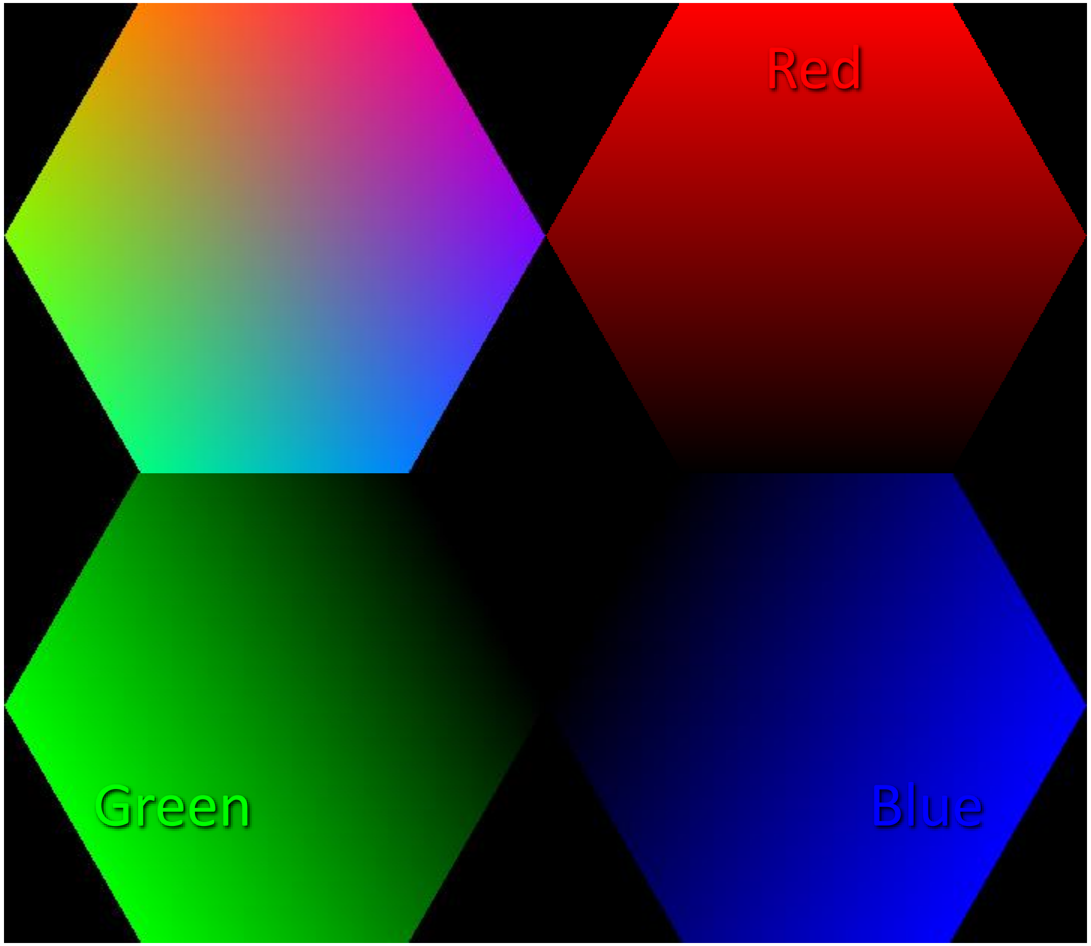


Here the hexagon is cut out of the cube to provide another view.

RGB Components of the Color Hexagon at Value 128

$r+g+b = 384$

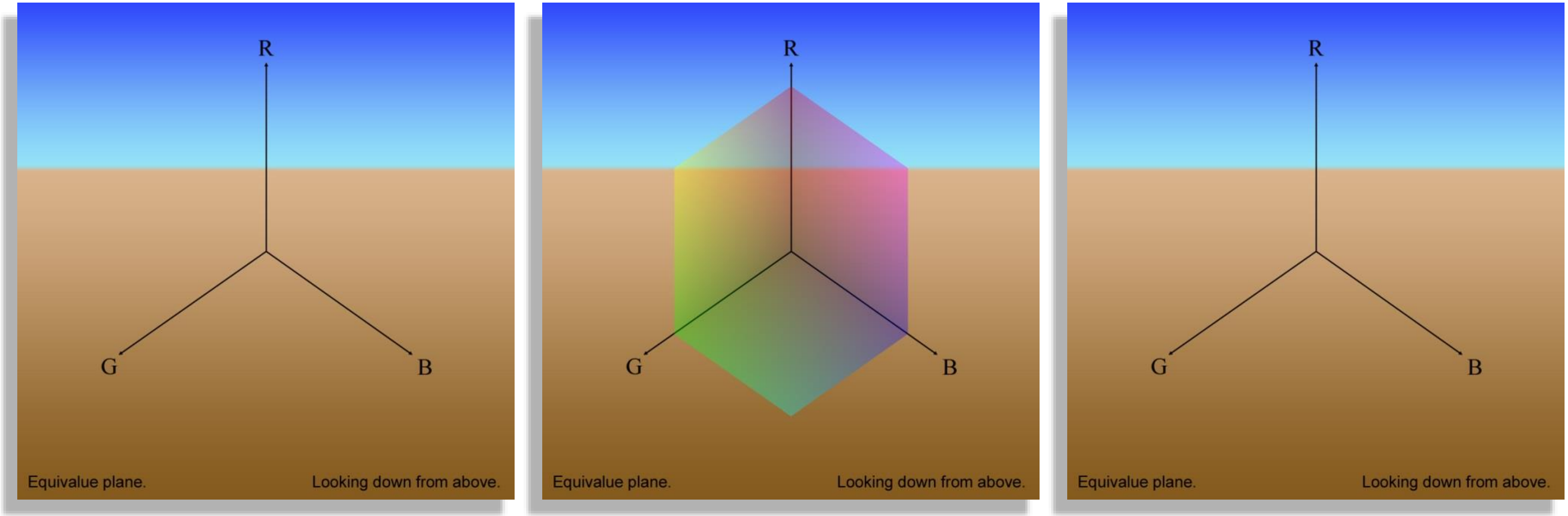
$r \in \{0,\ldots,255\}$



$g \in \{0,\ldots,255\}$

$b \in \{0,\ldots,255\}$

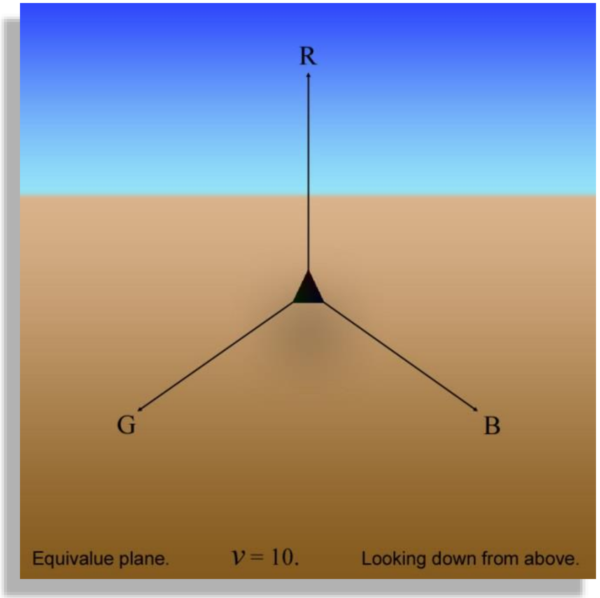
Equivalue Plane Intersecting Color Cube

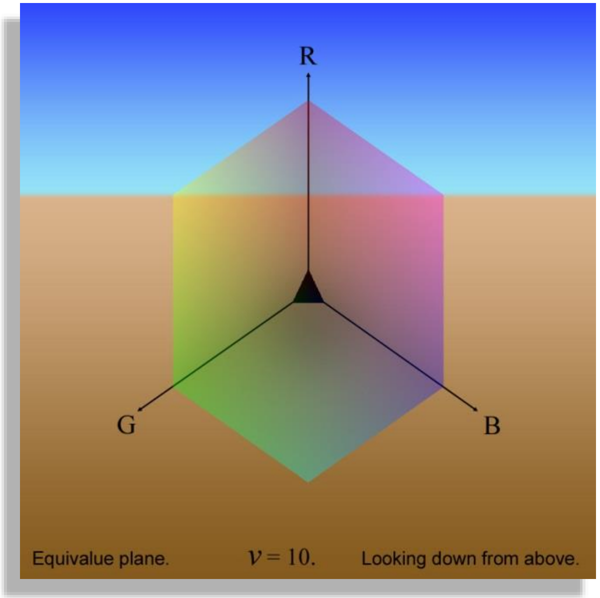


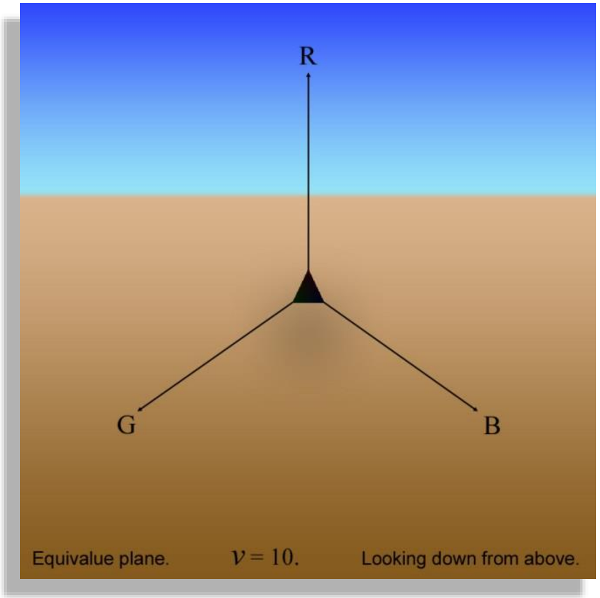
Projection: the gray line is perpendicular to this page.

Equivalue plane at $v = 0$: single point, pure black.

Equivalue Plane Intersecting Color Cube

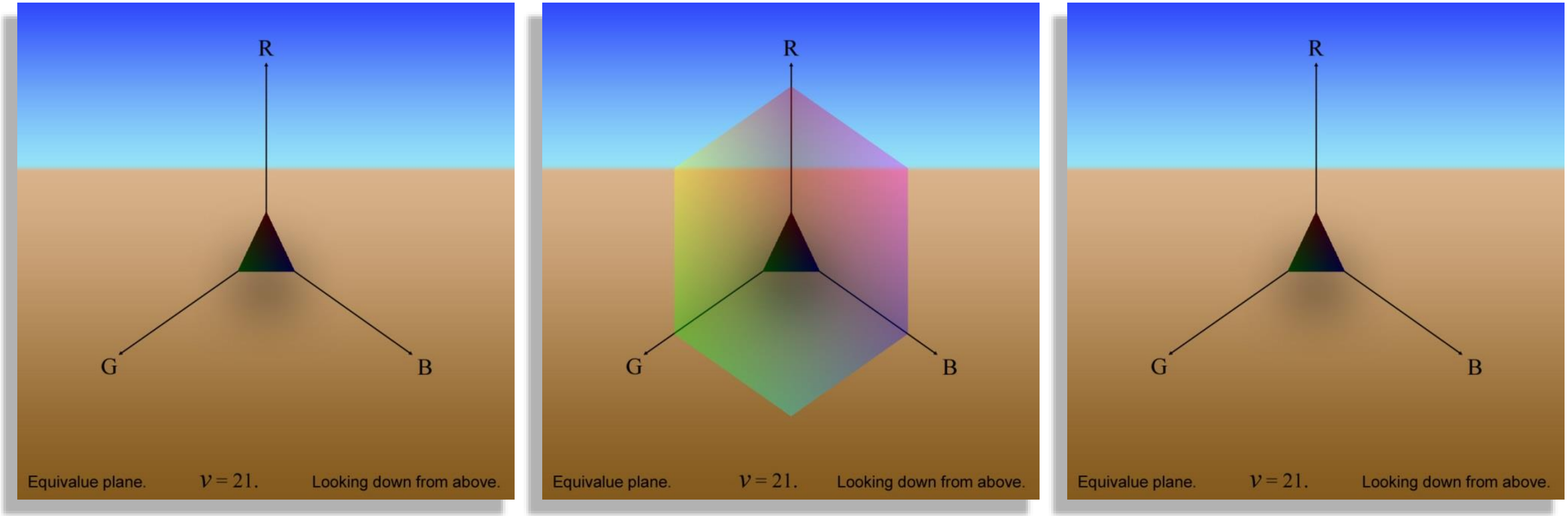






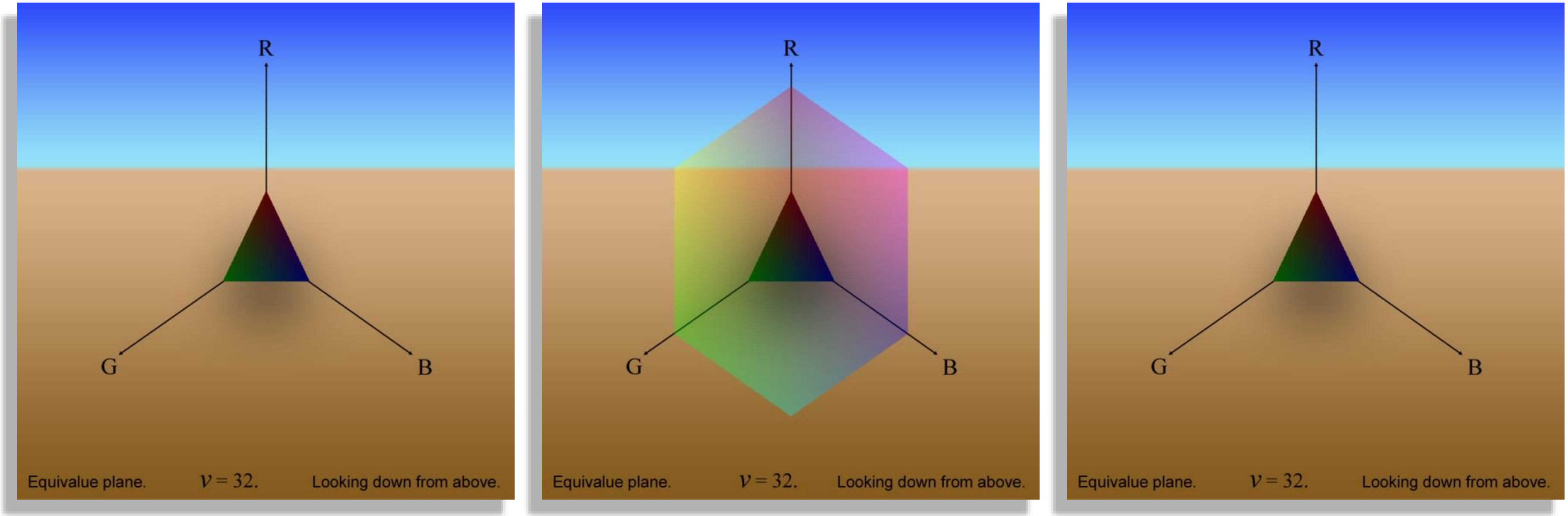
Projection: the gray line is perpendicular to this page.

Equivalue Plane Intersecting Color Cube



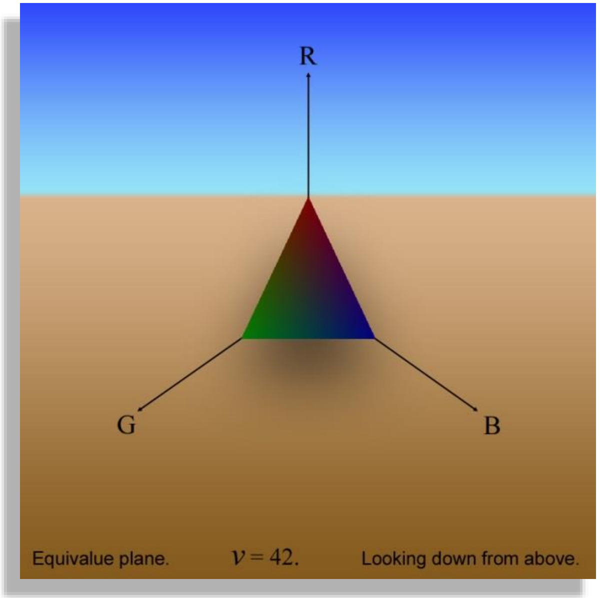
Projection: the gray line is perpendicular to this page.

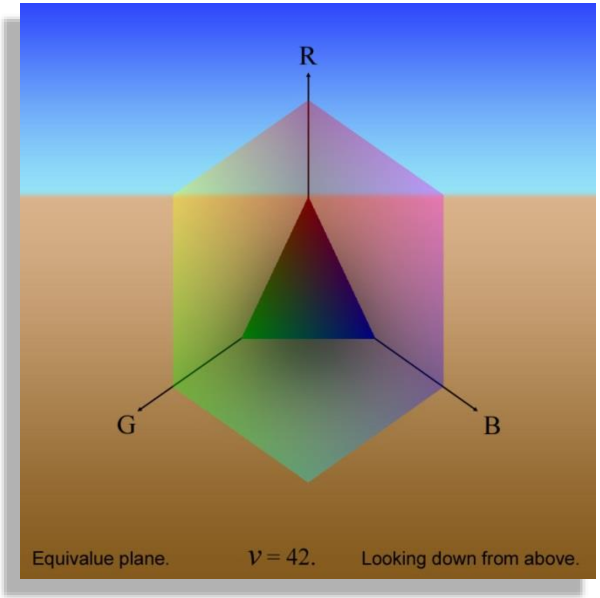
Equivalue Plane Intersecting Color Cube

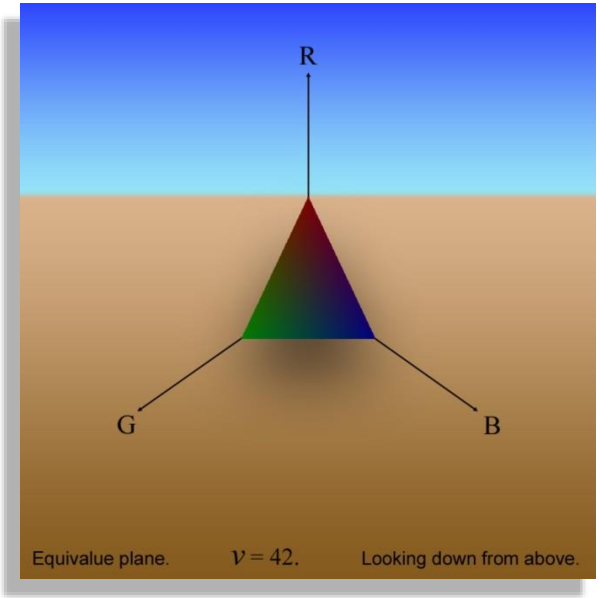


Projection: the gray line is perpendicular to this page.

Equivalue Plane Intersecting Color Cube

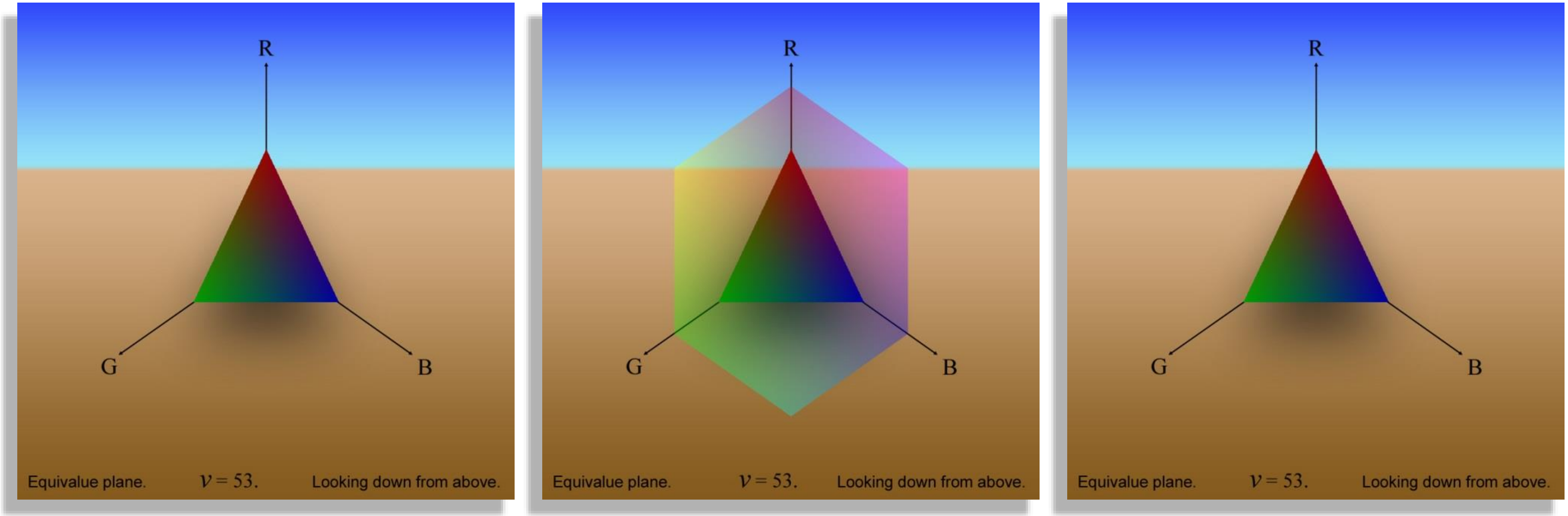






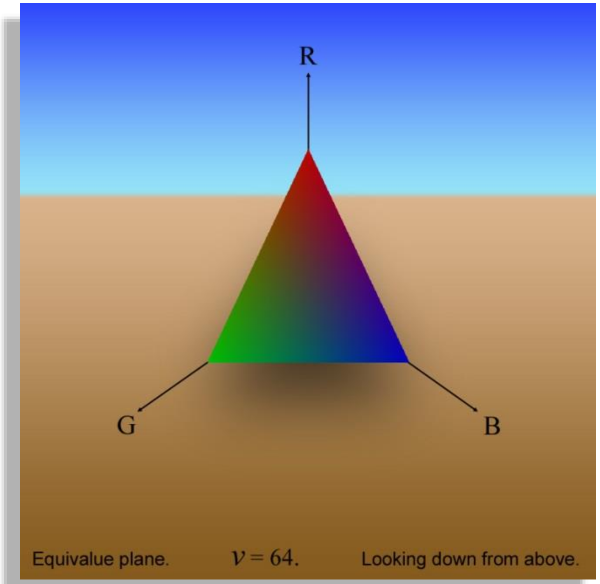
Projection: the gray line is perpendicular to this page.

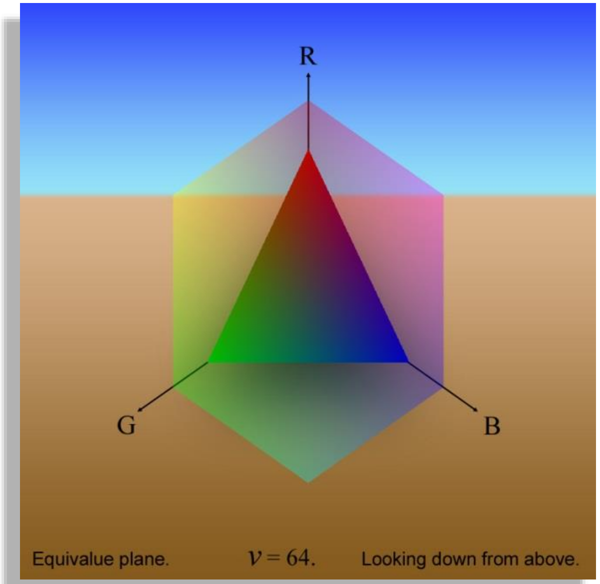
Equivalue Plane Intersecting Color Cube

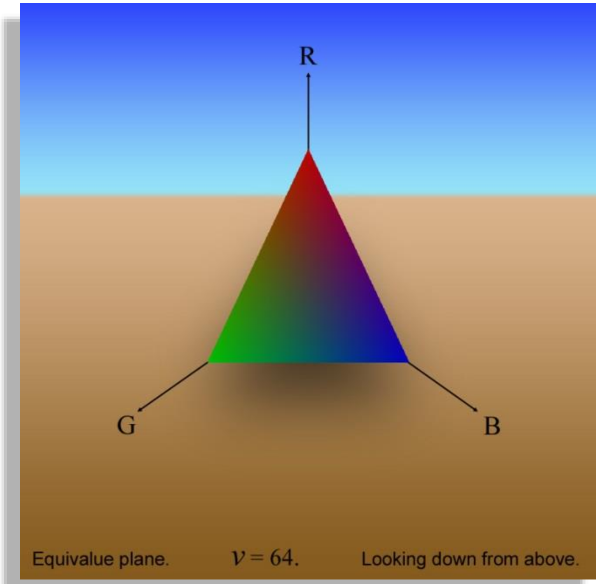


Projection: the gray line is perpendicular to this page.

Equivalue Plane Intersecting Color Cube

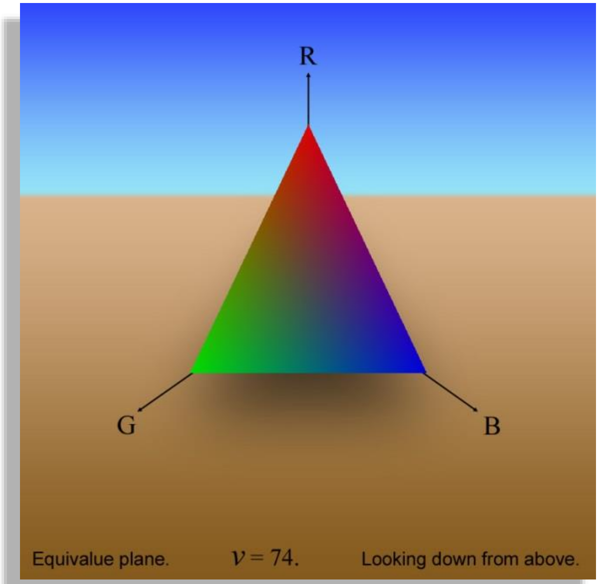


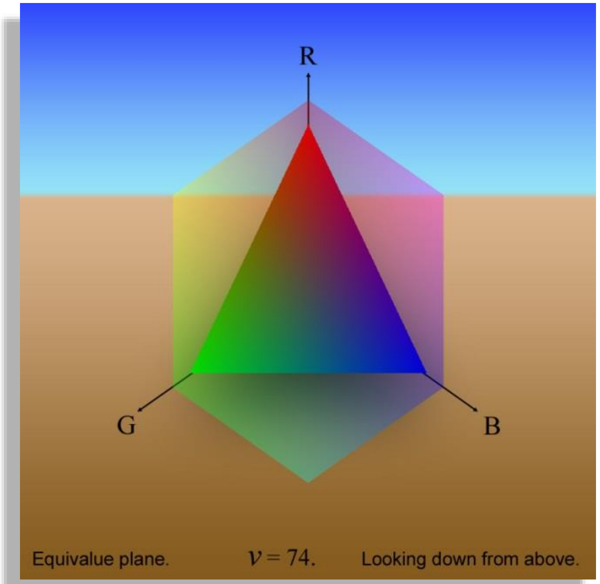


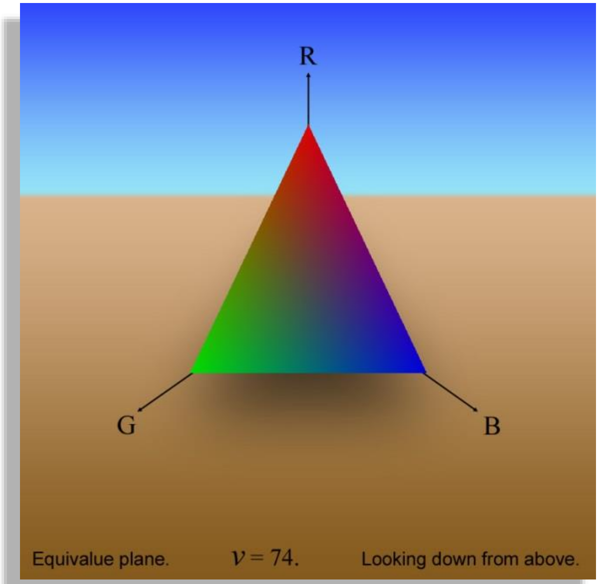


Projection: the gray line is perpendicular to this page.

Equivalue Plane Intersecting Color Cube

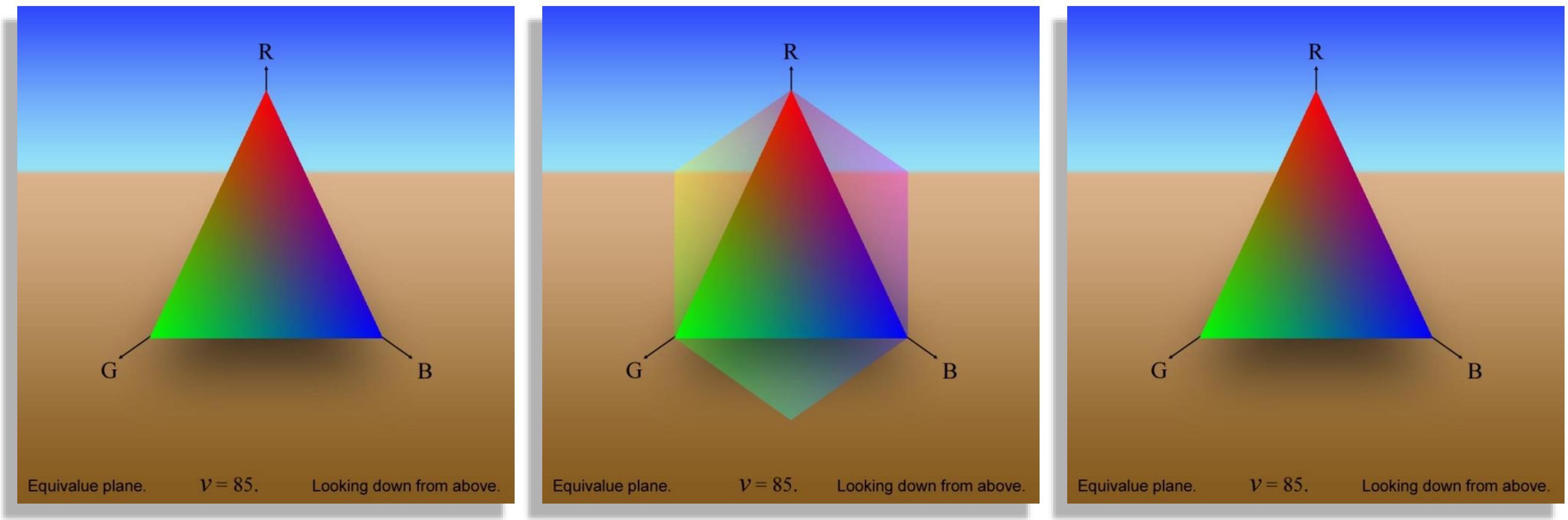






Projection: the gray line is perpendicular to this page.

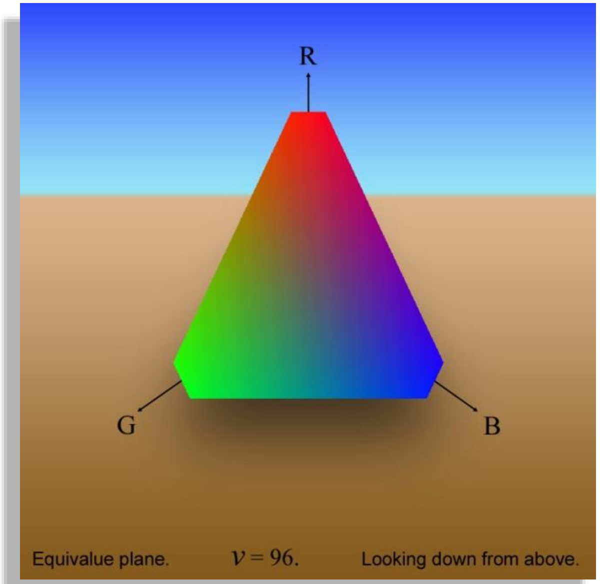
Equivalue Plane Intersecting Color Cube

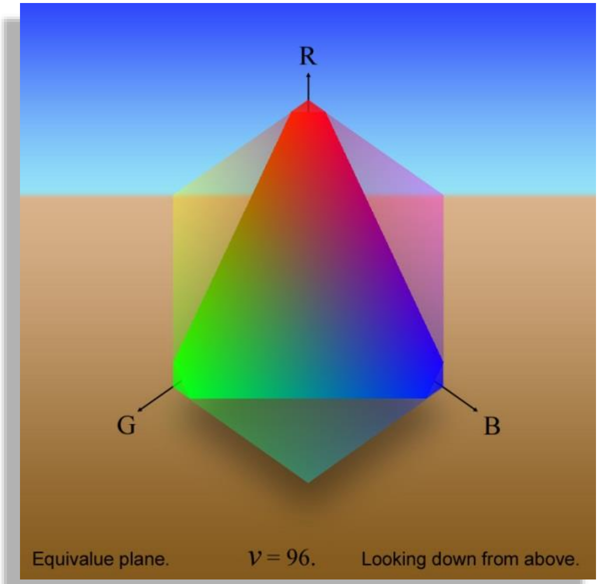


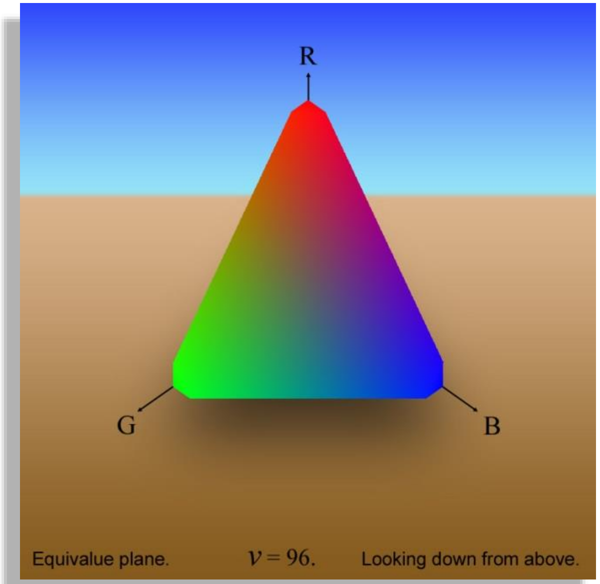
Projection: the gray line is perpendicular to this page.

Equivalue plane at $v = 85$: largest upright triangle, start of hexagonal intersections.

Equivalue Plane Intersecting Color Cube

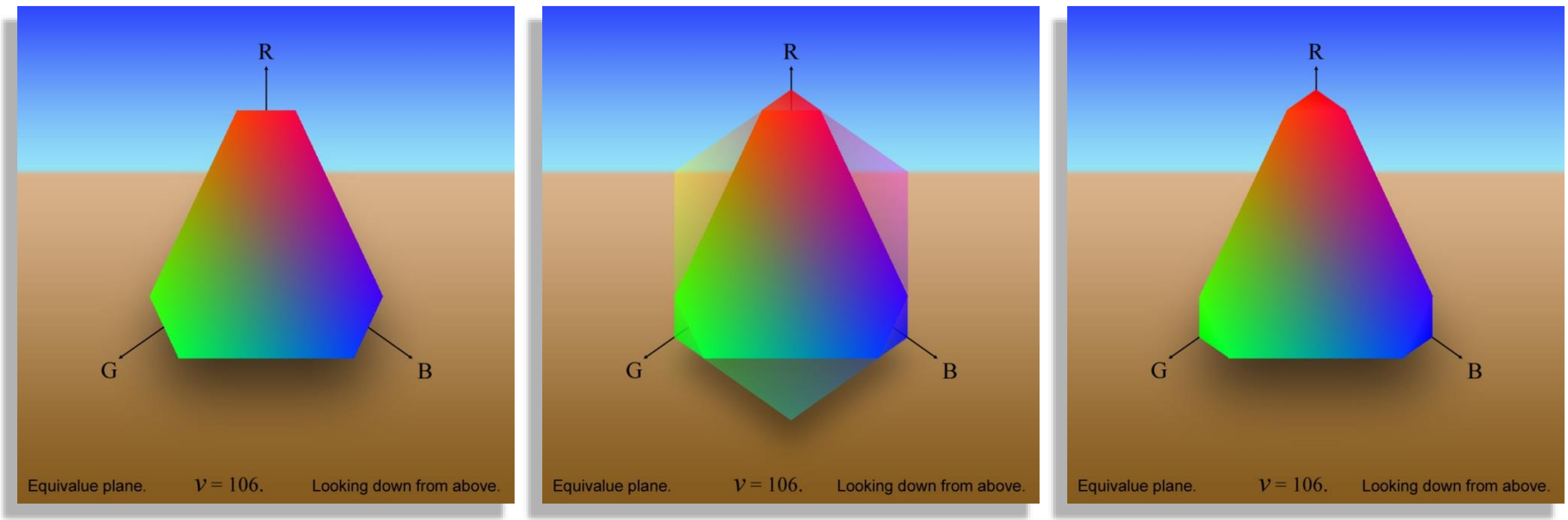






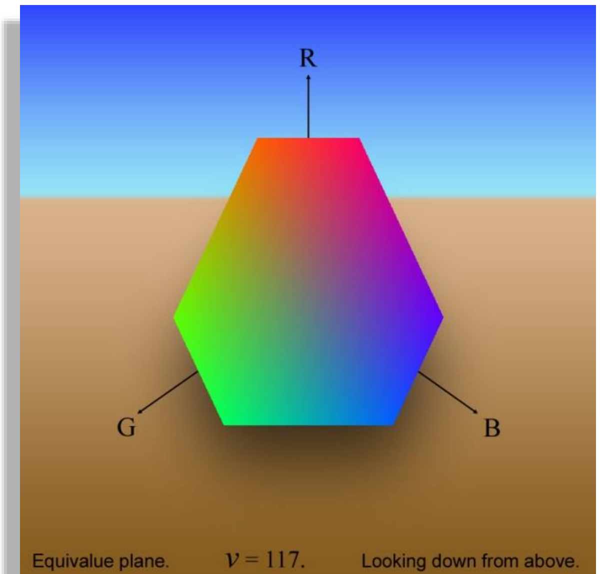
Projection: the gray line is perpendicular to this page.

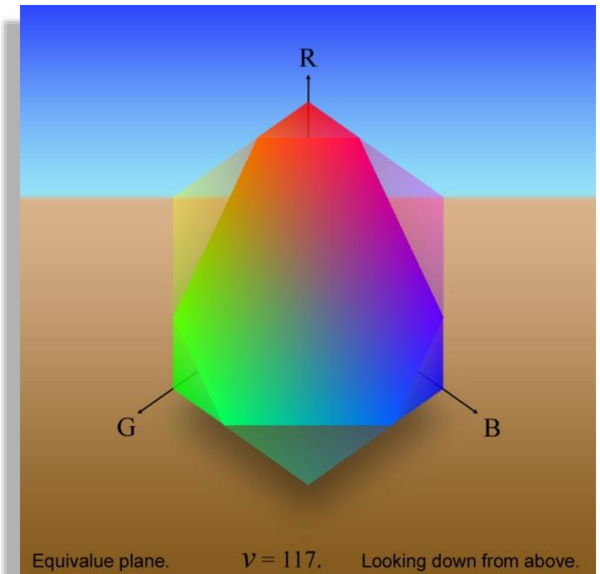
Equivalue Plane Intersecting Color Cube

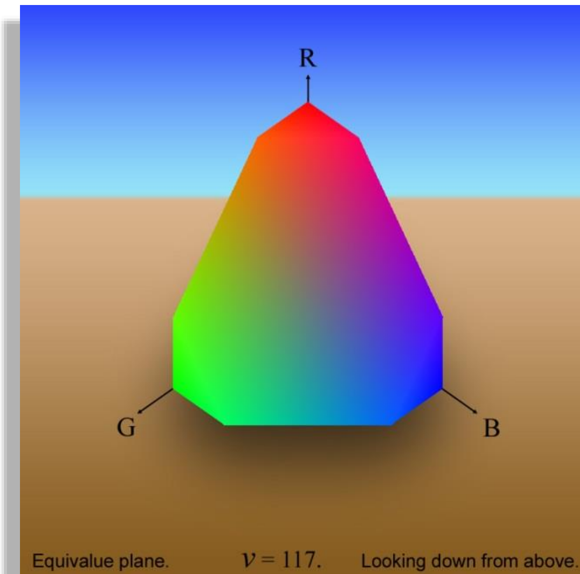


Projection: the gray line is perpendicular to this page.

Equivalue Plane Intersecting Color Cube

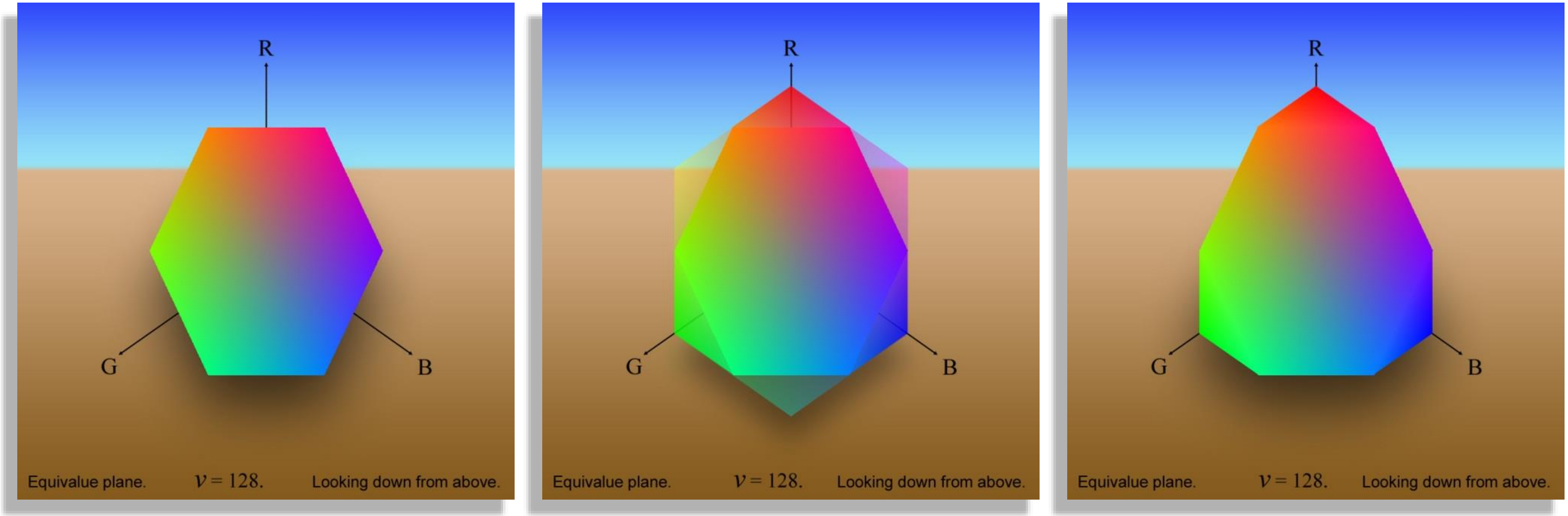






Projection: the gray line is perpendicular to this page.

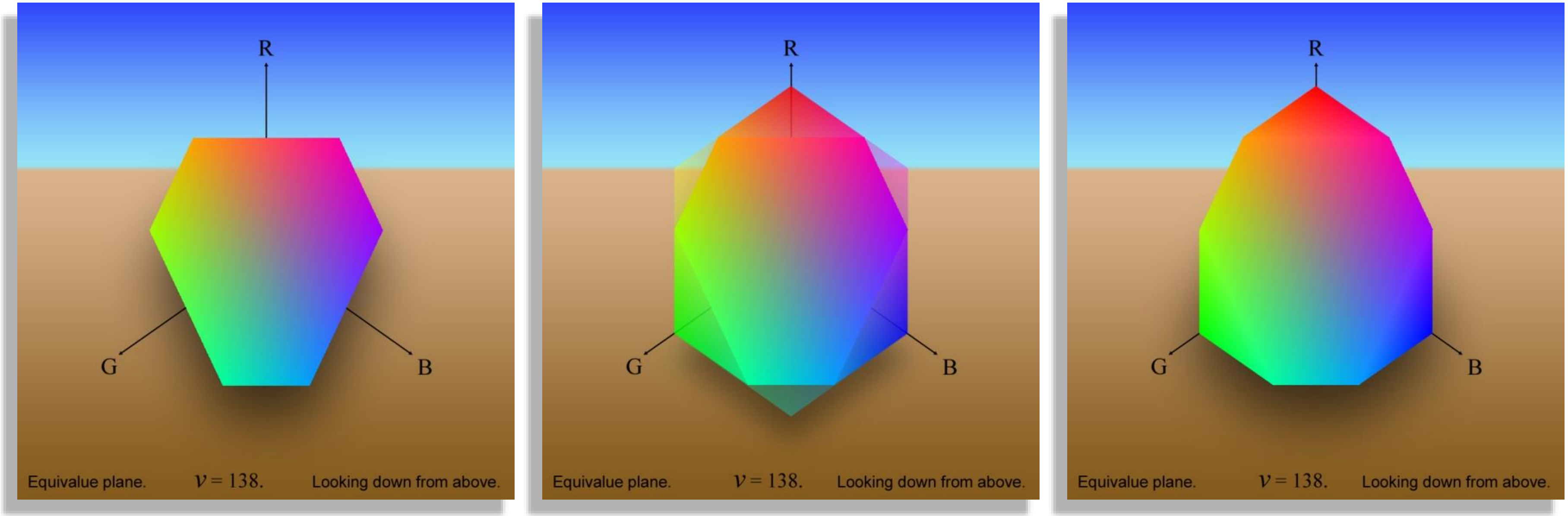
Equivalue Plane Intersecting Color Cube



Projection: the gray line is perpendicular to this page.

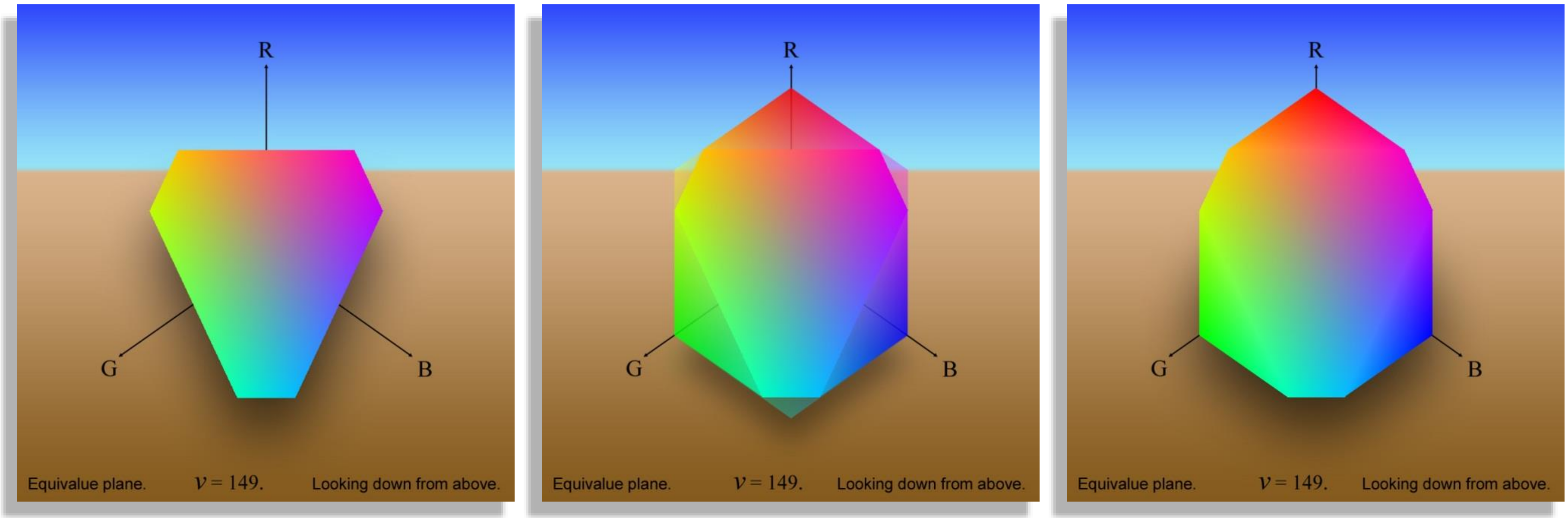
Equivalue plane at $v = 128$: symmetric hexagon, intersecting plane with largest area.

Equivalue Plane Intersecting Color Cube



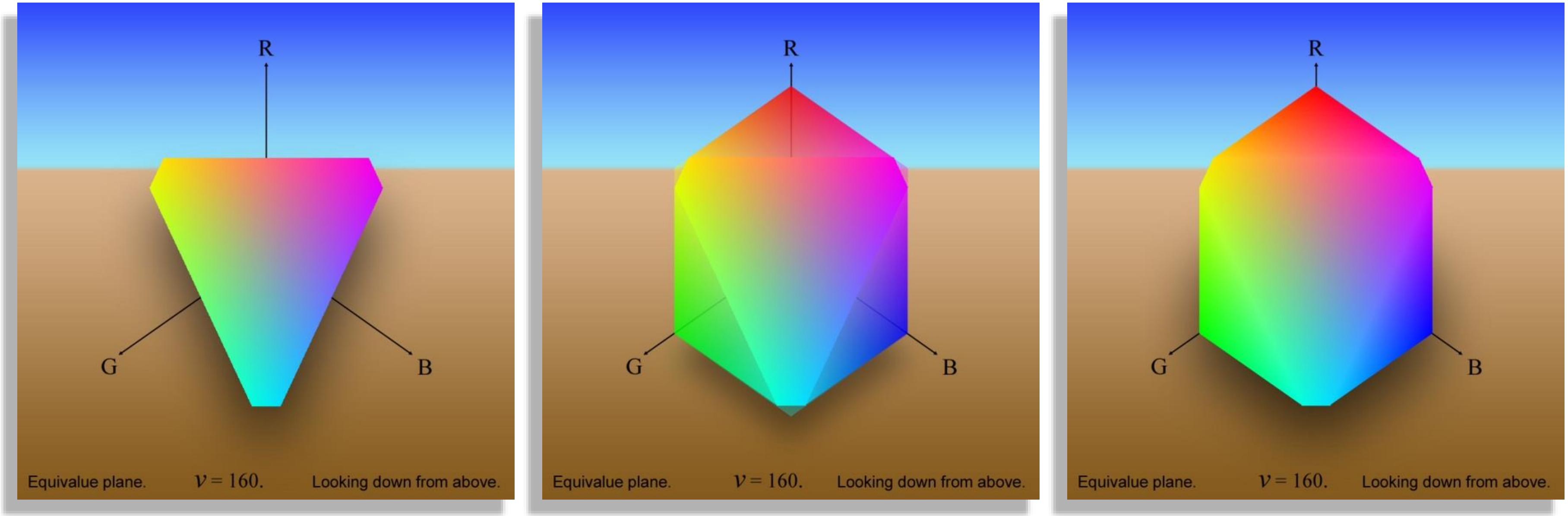
Projection: the gray line is perpendicular to this page.

Equivalue Plane Intersecting Color Cube



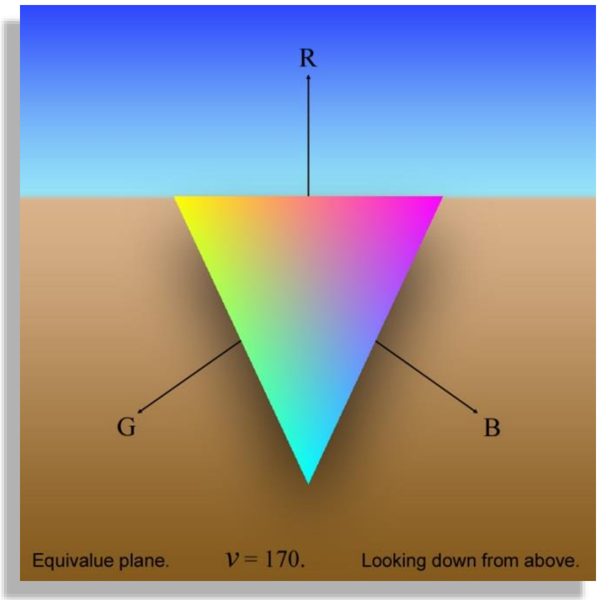
Projection: the gray line is perpendicular to this page.

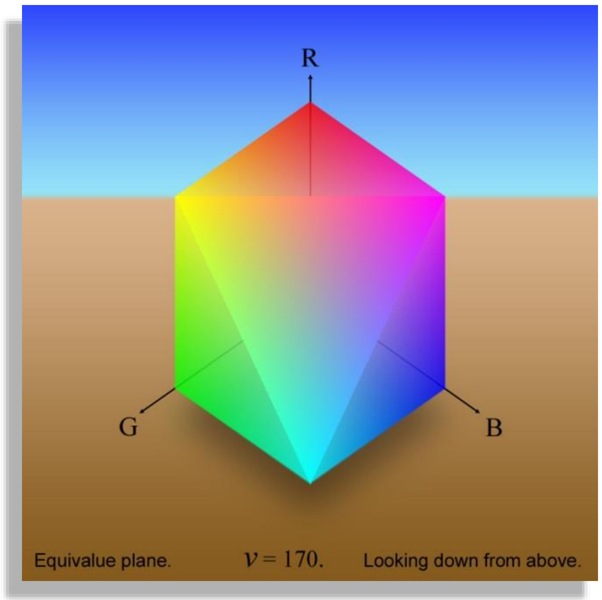
Equivalue Plane Intersecting Color Cube

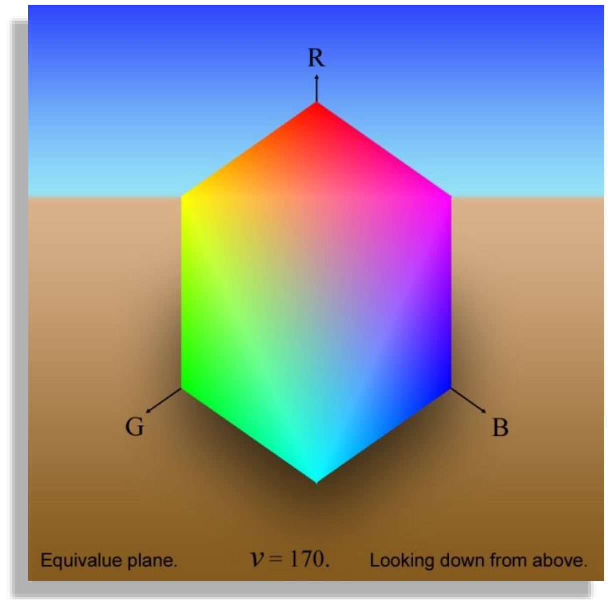


Projection: the gray line is perpendicular to this page.

Equivalue Plane Intersecting Color Cube



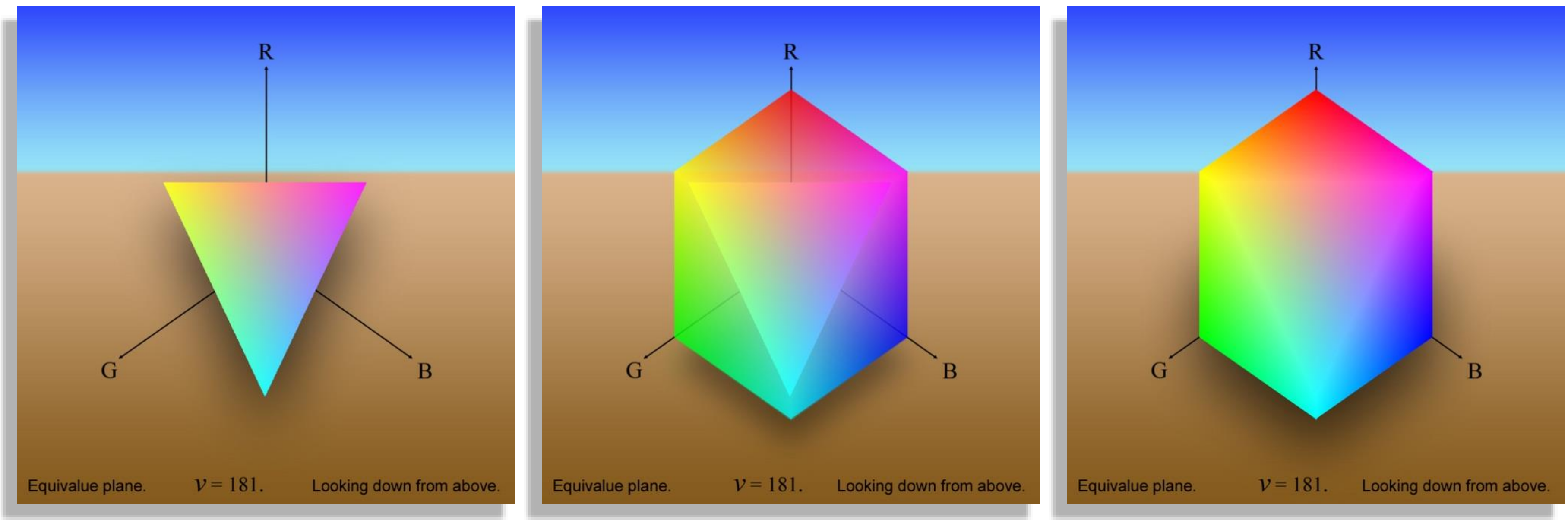




Projection: the gray line is perpendicular to this page.

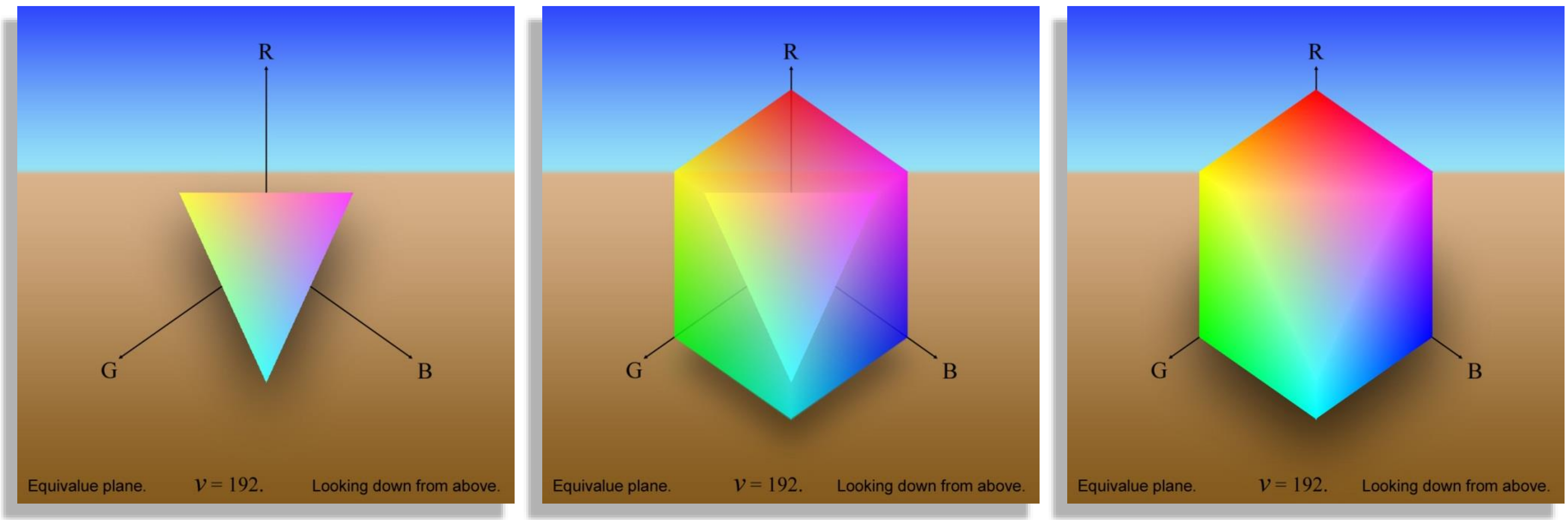
Equivalue plane at $v = 170$: largest inverted triangle, end of hexagonal intersections.

Equivalue Plane Intersecting Color Cube



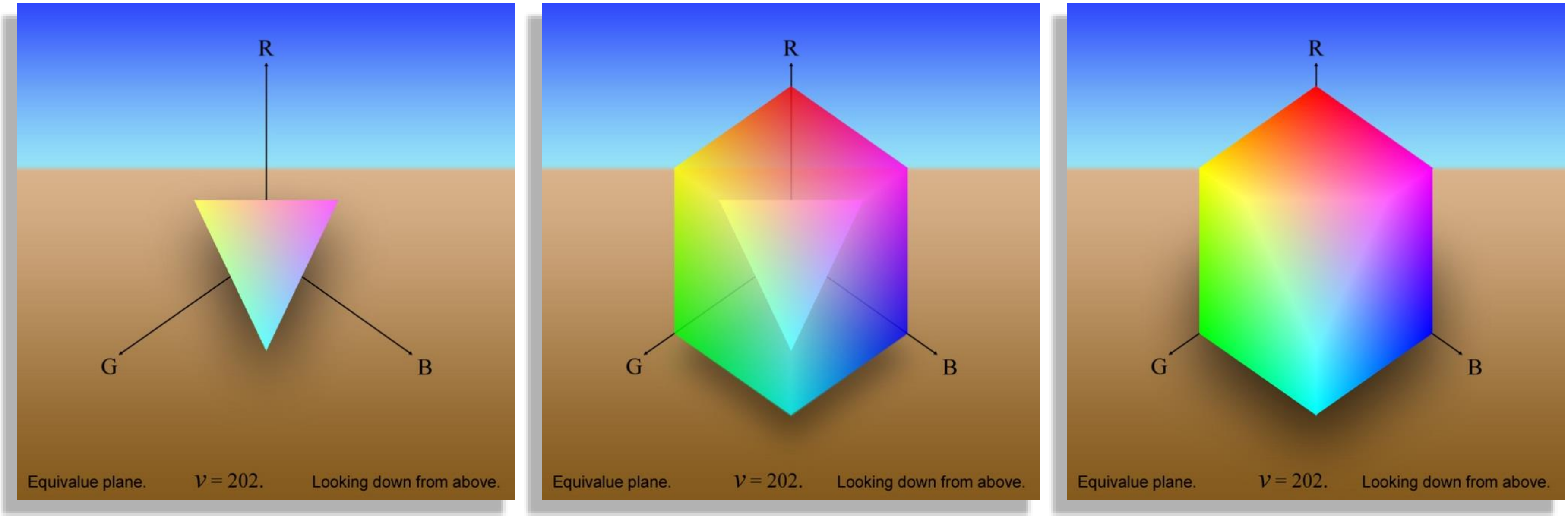
Projection: the gray line is perpendicular to this page.

Equivalue Plane Intersecting Color Cube



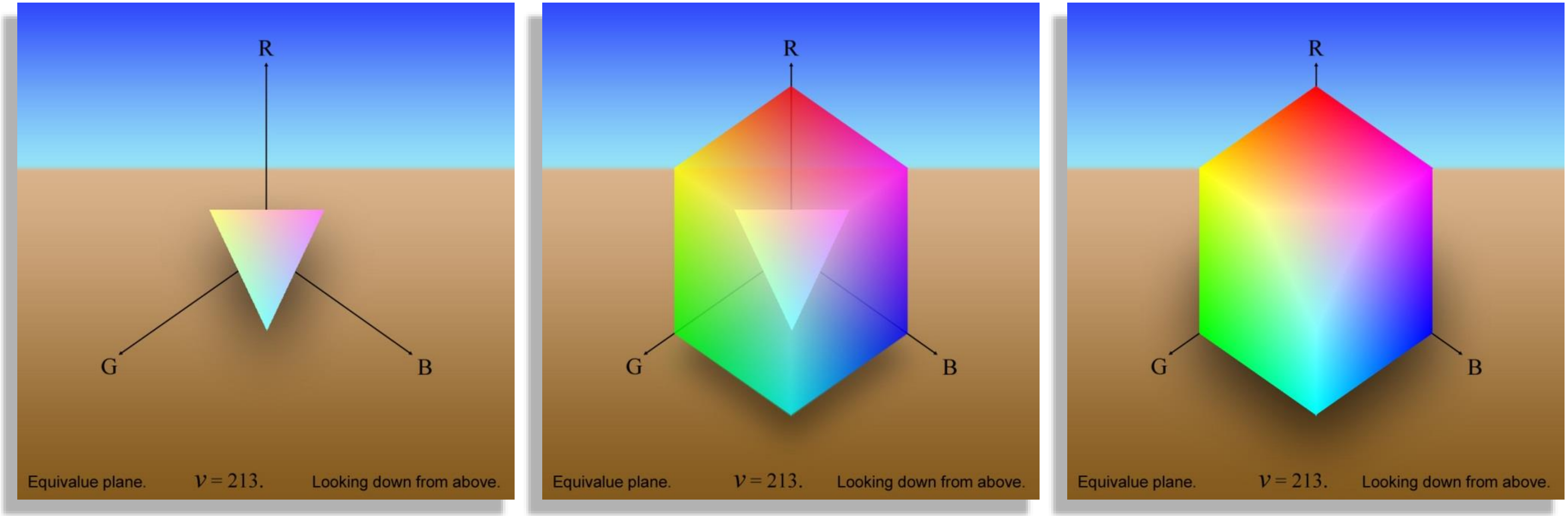
Projection: the gray line is perpendicular to this page.

Equivalue Plane Intersecting Color Cube



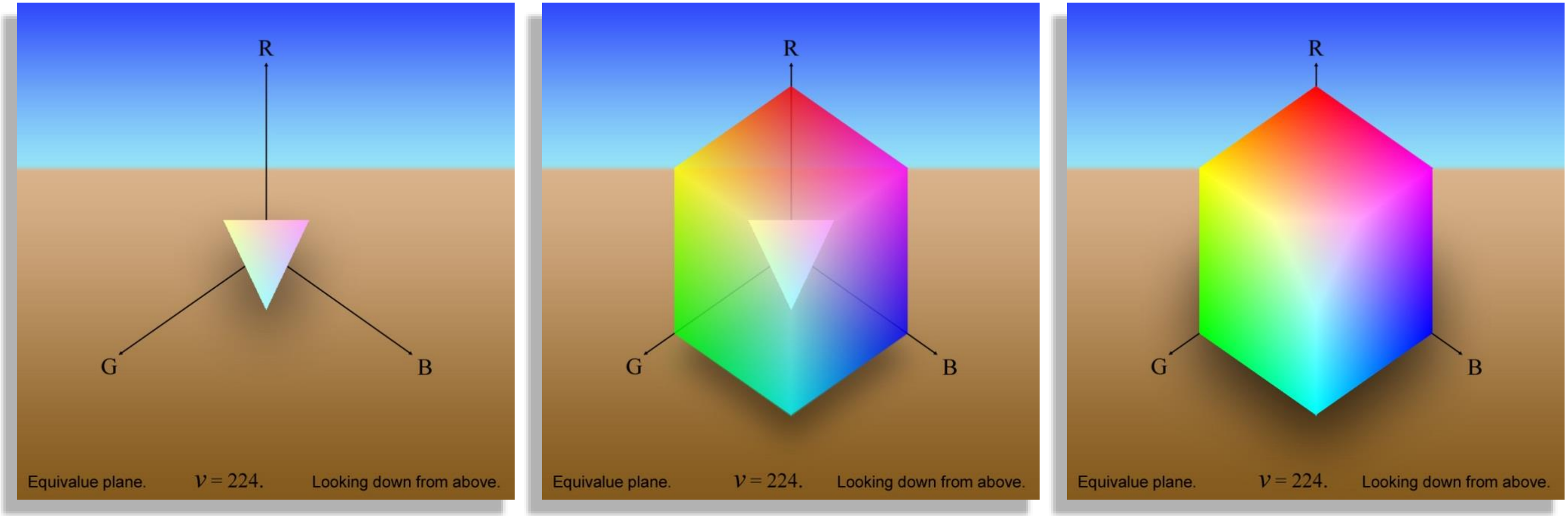
Projection: the gray line is perpendicular to this page.

Equivalue Plane Intersecting Color Cube



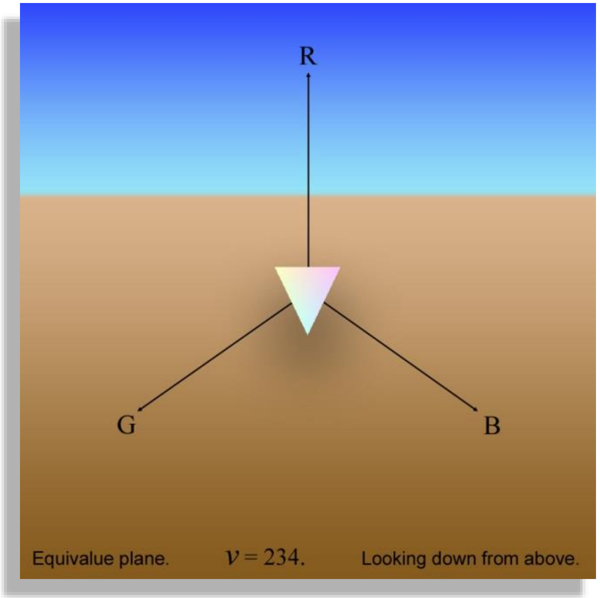
Projection: the gray line is perpendicular to this page.

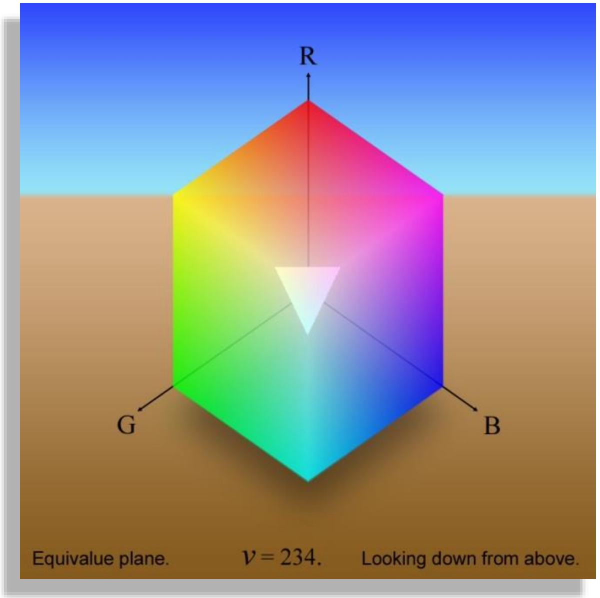
Equivalue Plane Intersecting Color Cube

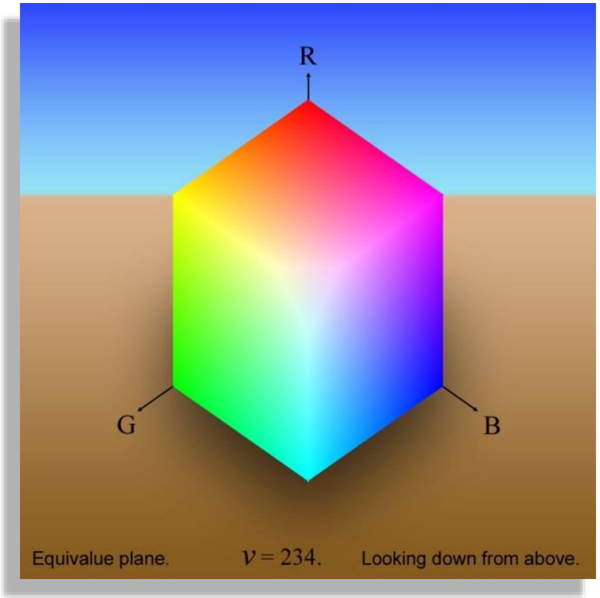


Projection: the gray line is perpendicular to this page.

Equivalue Plane Intersecting Color Cube

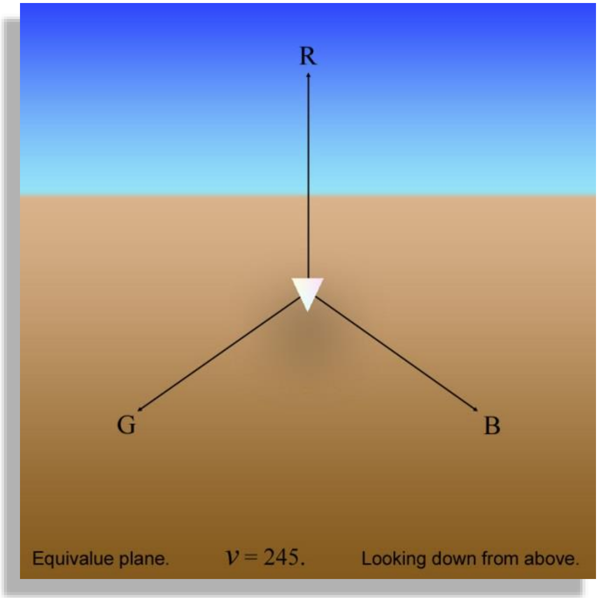


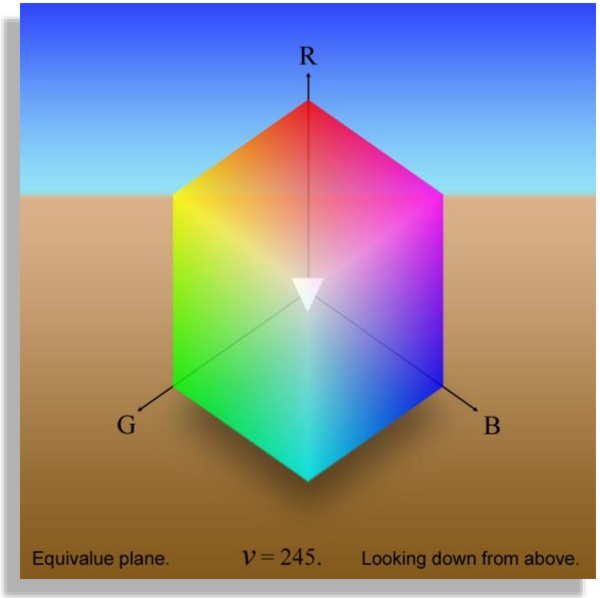


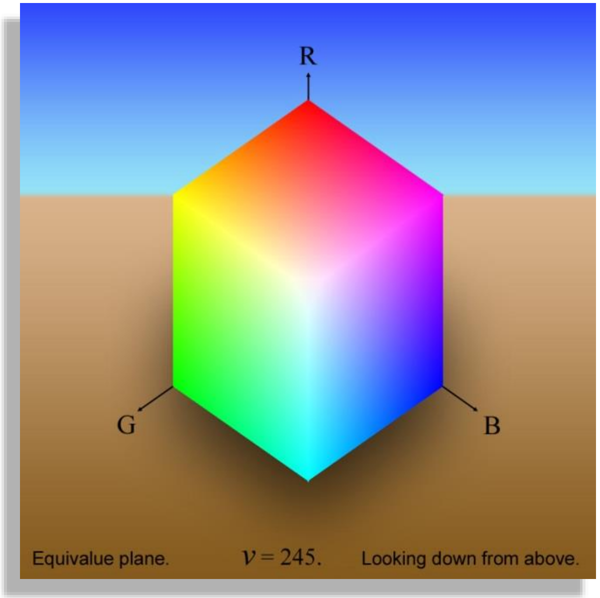


Projection: the gray line is perpendicular to this page.

Equivalue Plane Intersecting Color Cube

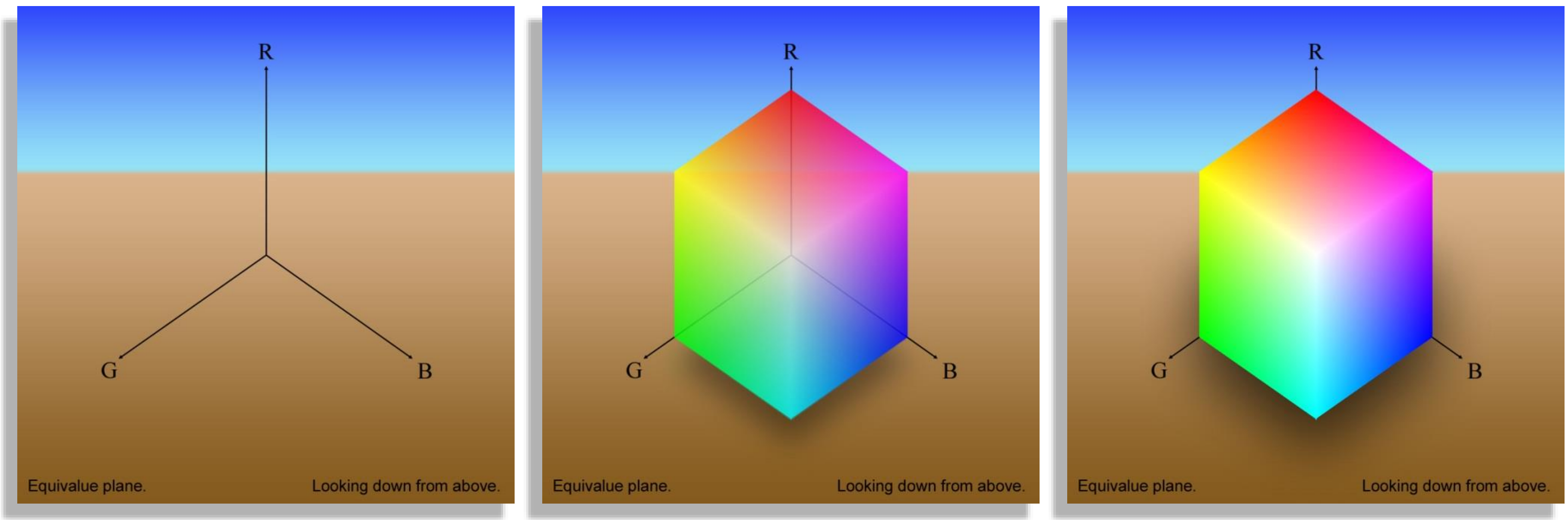






Projection: the gray line is perpendicular to this page.

Equivalue Plane Intersecting Color Cube



Projection: the gray line is perpendicular to this page.

Equivalue plane at $v = 255$: single point, pure white.

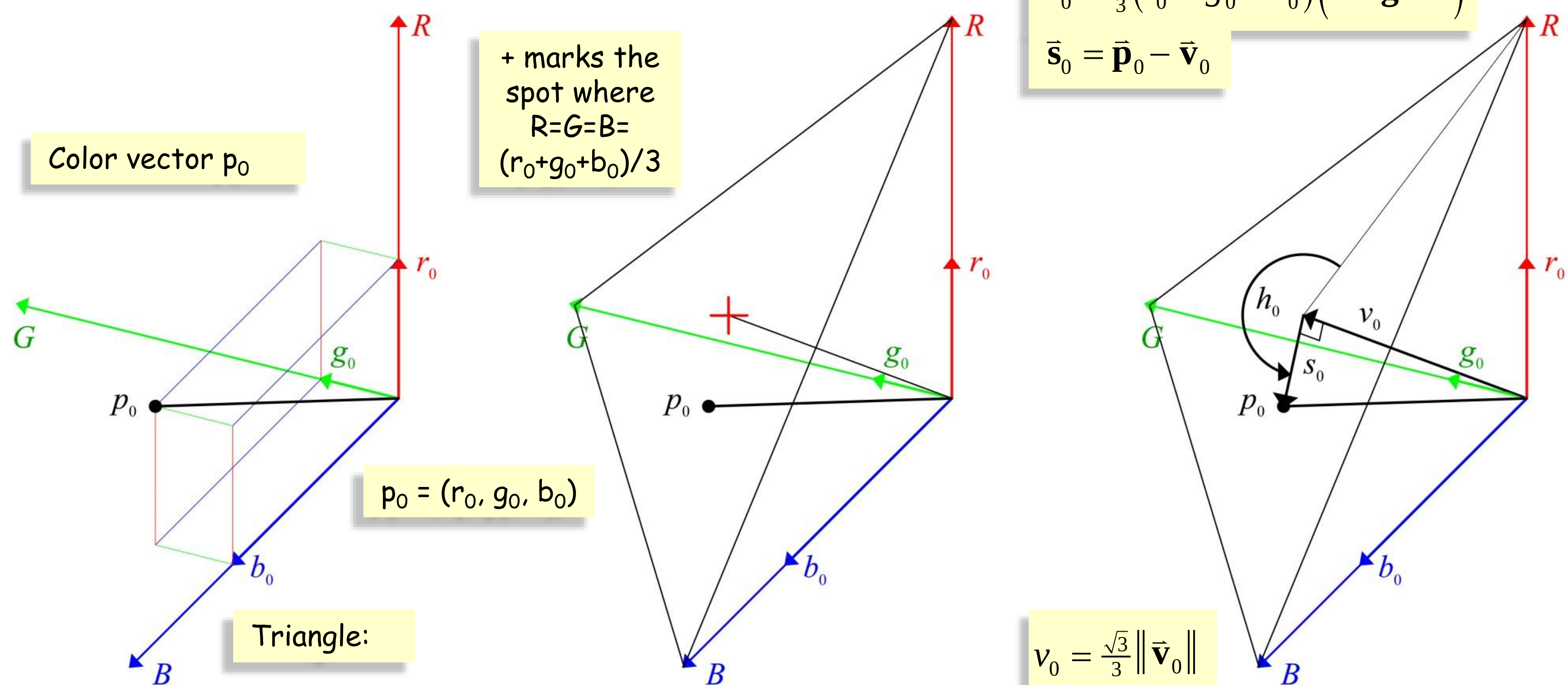
HSV Color Representation

$$\vec{\mathbf{v}}_0 = \tfrac{1}{3}\left(r_0 + g_0 + b_0\right)\left(\hat{\mathbf{r}} + \hat{\mathbf{g}} + \hat{\mathbf{b}}\right)$$

$$\vec{\mathbf{s}}_0 = \vec{\mathbf{p}}_0 - \vec{\mathbf{v}}_0$$

Color vector p_0

+ marks the spot where R=G=B= $(r_0+g_0+b_0)/3$



$p_0 = (r_0, g_0, b_0)$

Triangle:

lies in plane $R+G+B=(r_0+g_0+b_0)$,

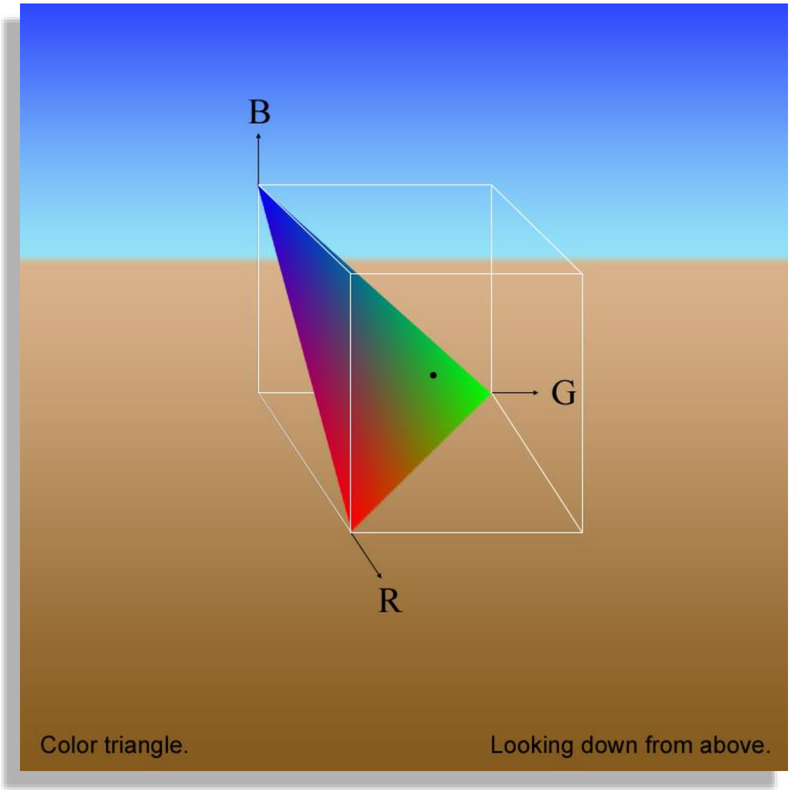
intersects R, G, & B axes at $r_0+g_0+b_0$, and contains p_0.

$$v_0 = \tfrac{\sqrt{3}}{3}\left\|\vec{\mathbf{v}}_0\right\|$$

$$s_0 = \left\|\vec{\mathbf{s}}_0\right\|$$

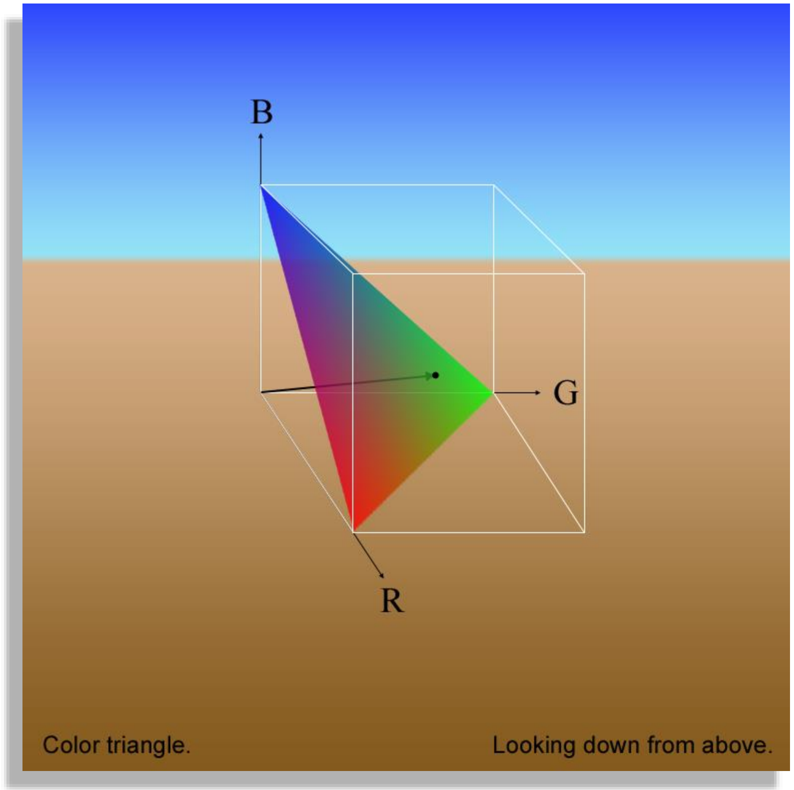
$$h_0 = \angle\left\{\left[\left(r_0 + g_0 + b_0\right)\hat{\mathbf{r}} - \vec{\mathbf{v}}_0\right], \vec{\mathbf{s}}_0\right\}$$

A Color Point on its Equivalue Plane*

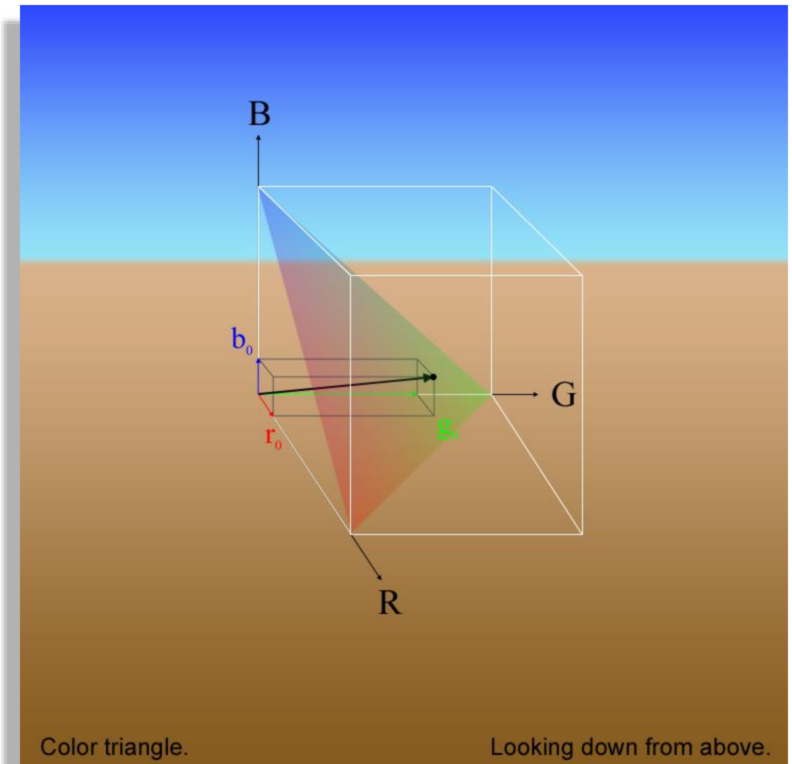


*where all the colors have components r, g, & b such that r+g+b = the same constant c.

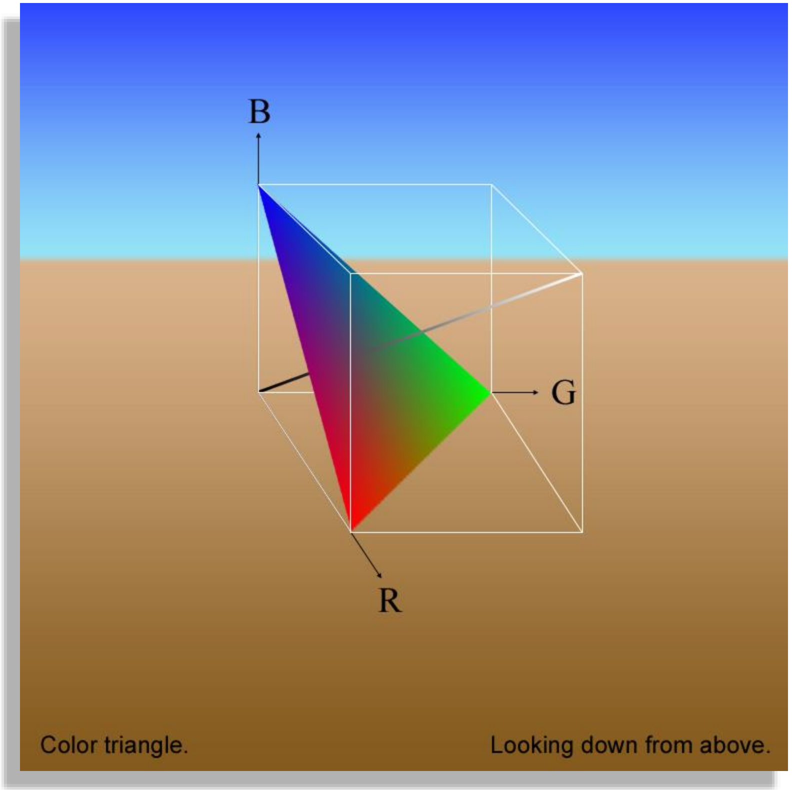
The Color Vector Associated with the Color Point



The Vector's Coordinates and RGB Component Vectors

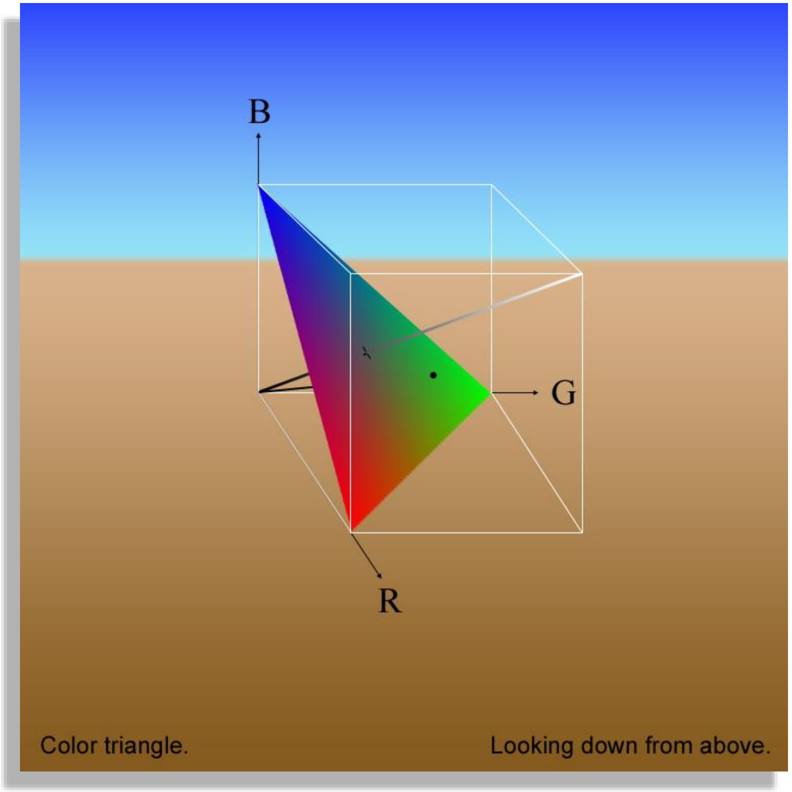


The Color Cube, the Equivalue Plane, & the Gray Line*



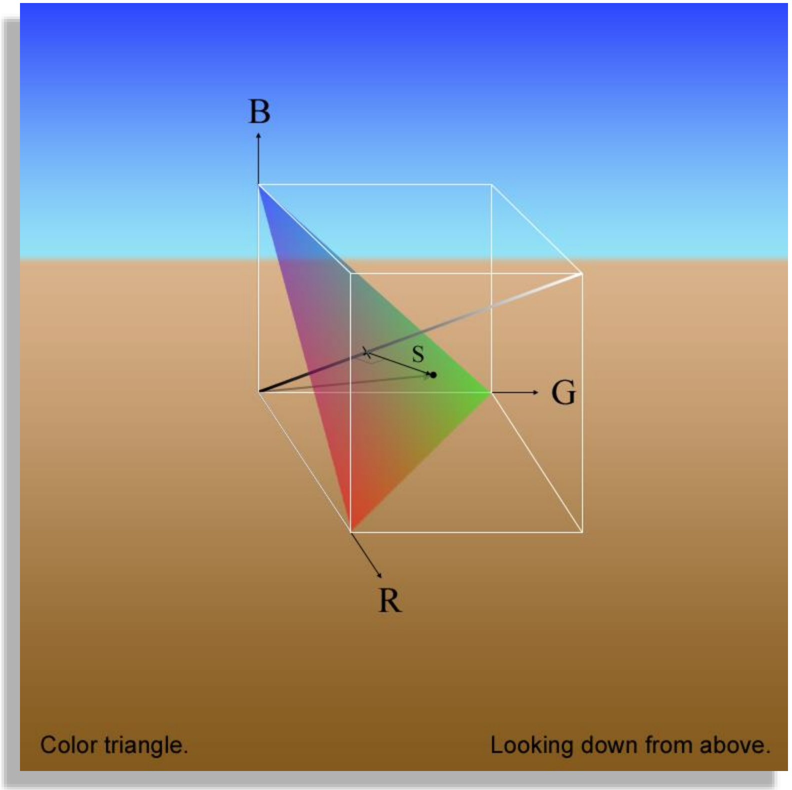
*The point at which the gray line intersects equivalue plane is called the “gray point” .

The Color Point and the Gray Line*



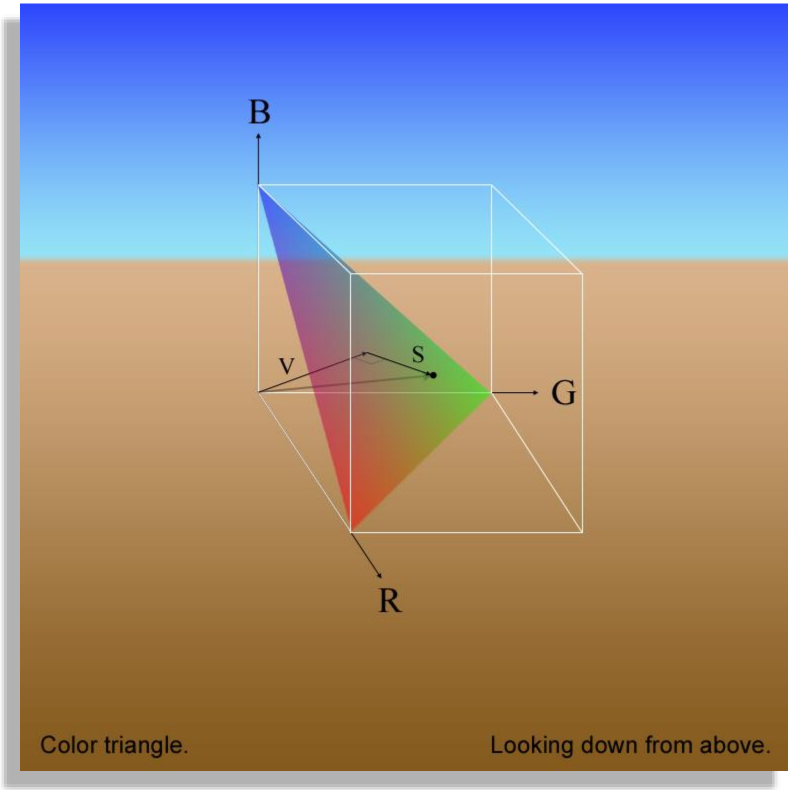
*"+" marks the gray point on the equivalue plane.

The Saturation Component* of the Color Vector

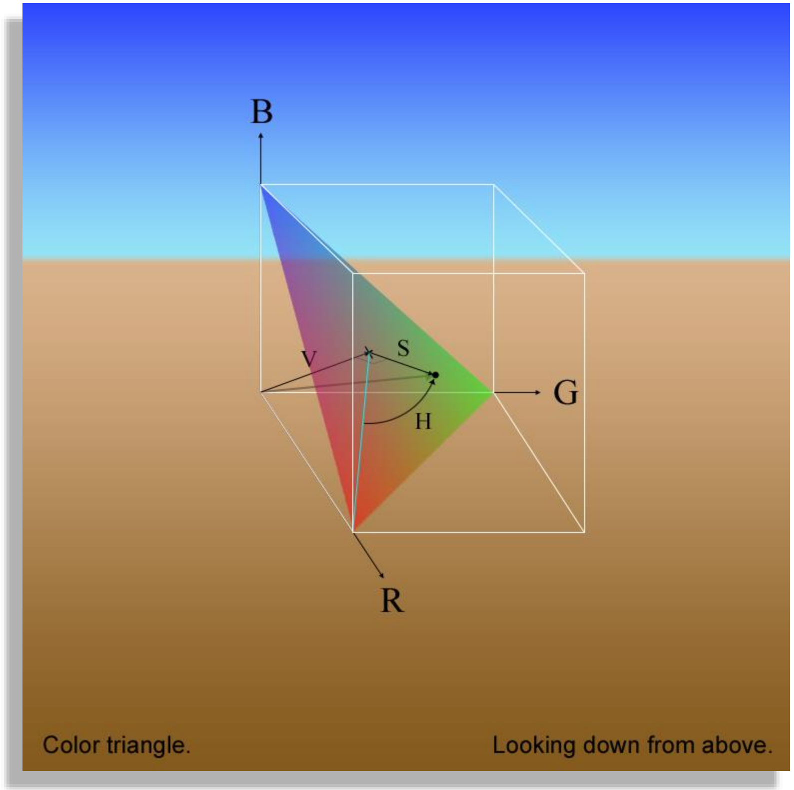


*is a vector in the equivalue plane from the gray point to the color point; its length is the scalar saturation.

The Saturation and Value Components of the Color Vector

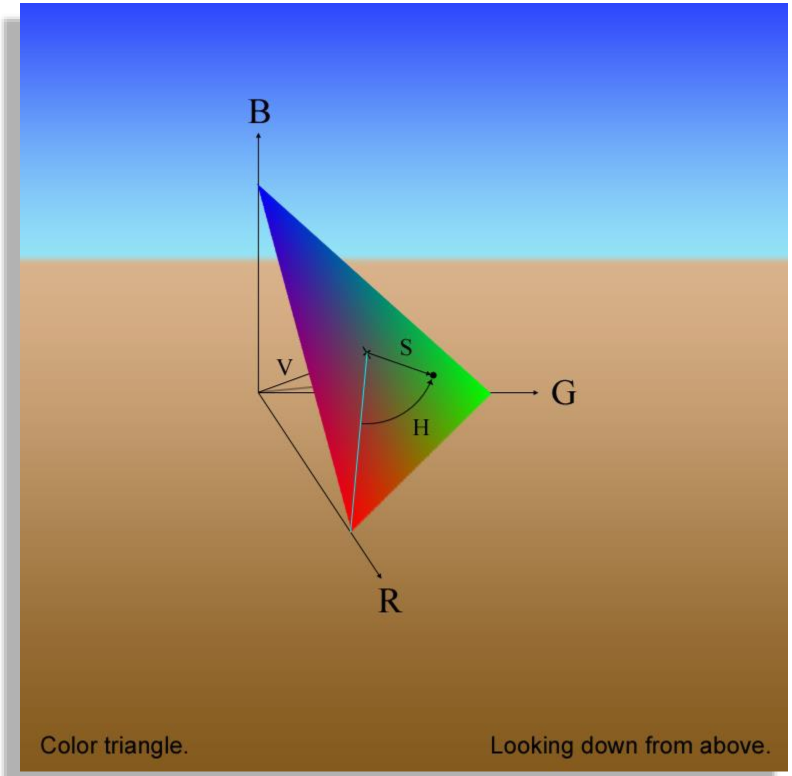


The Hue*, Saturation, and Value of the Color

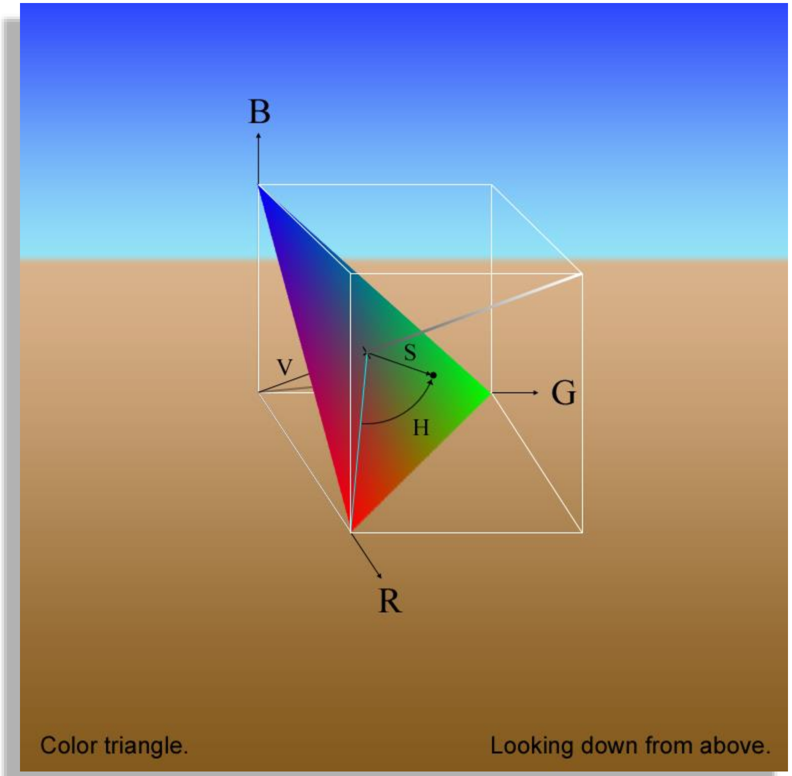


*is the angle in the plane from the red line (in the plane from the gray point to the red axis) to the color point.

The Hue and the Saturation on the Equivalue Plane



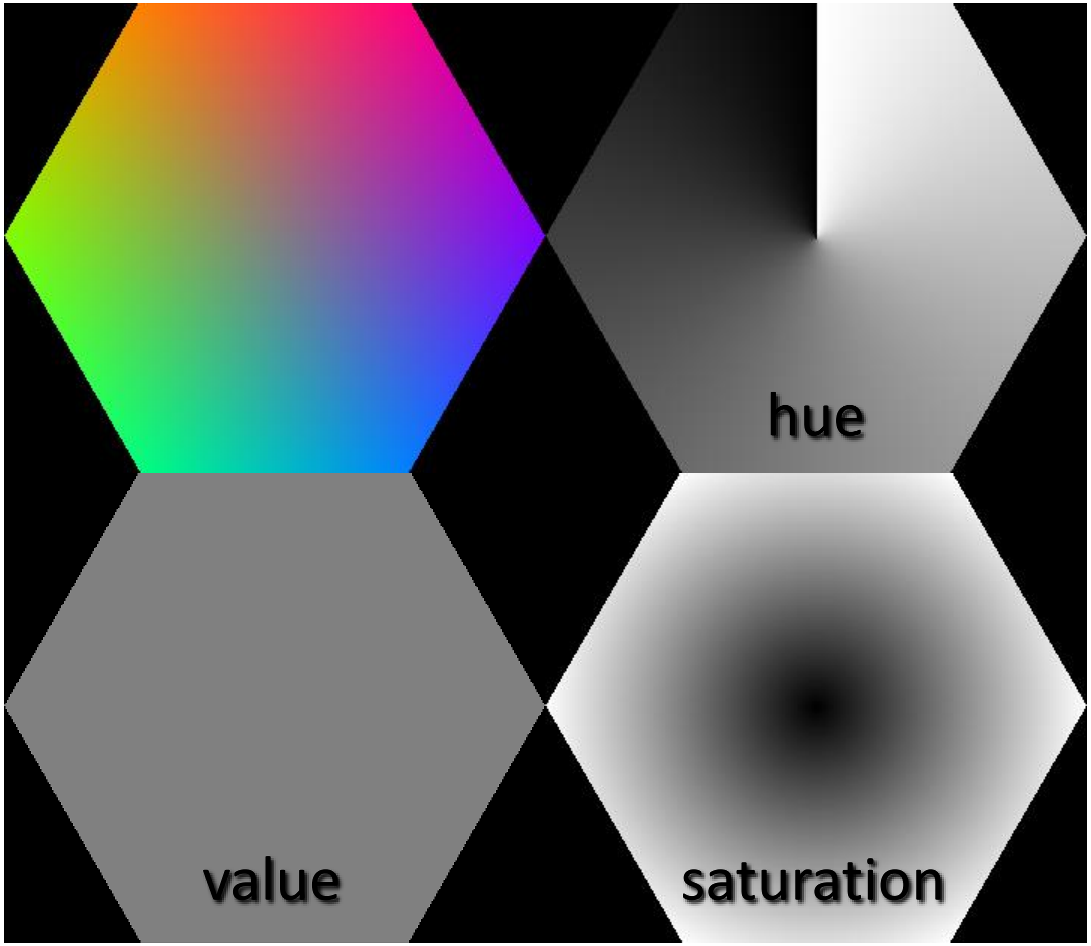
The Hue, Saturation, and Value with the Gray Line



HSV Components of the Color Hexagon at Value 128

r+g+b = 384

v = 128

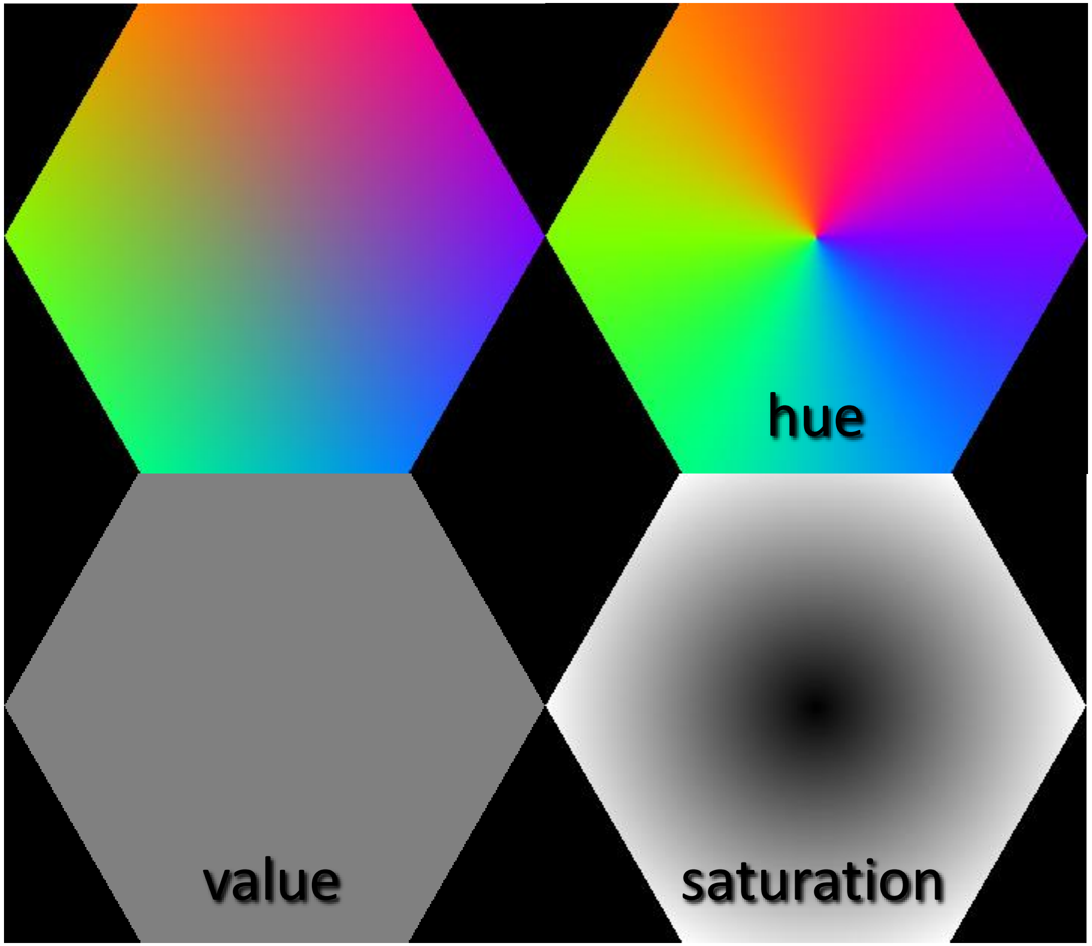


$h \in (\pi/128) \times \{0,\ldots,255\}$

$s \in (1/255) \times \{0,\ldots,255\}$

HSV Components of the Color Hexagon at Value 128

r+g+b = 384



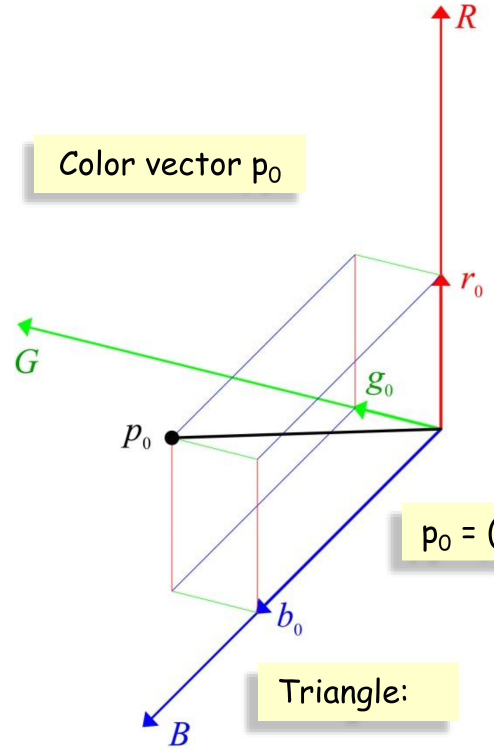
$h \in [0,2\pi)$

v = 128

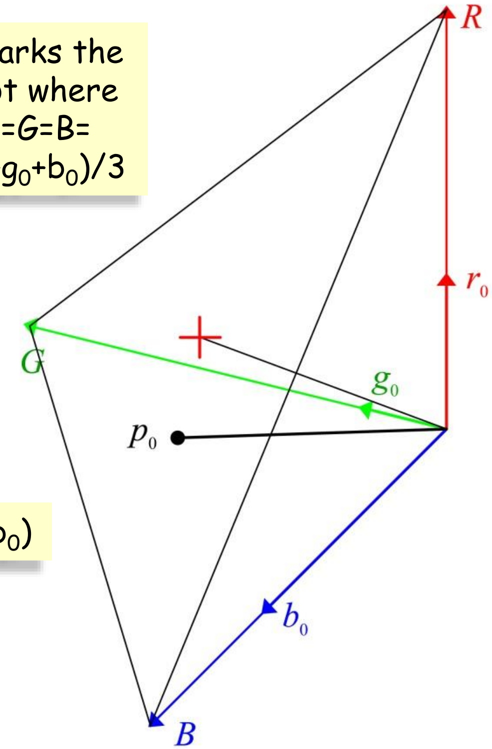
$s \in [0,1]$

HSV Color Representation

Color vector p_0

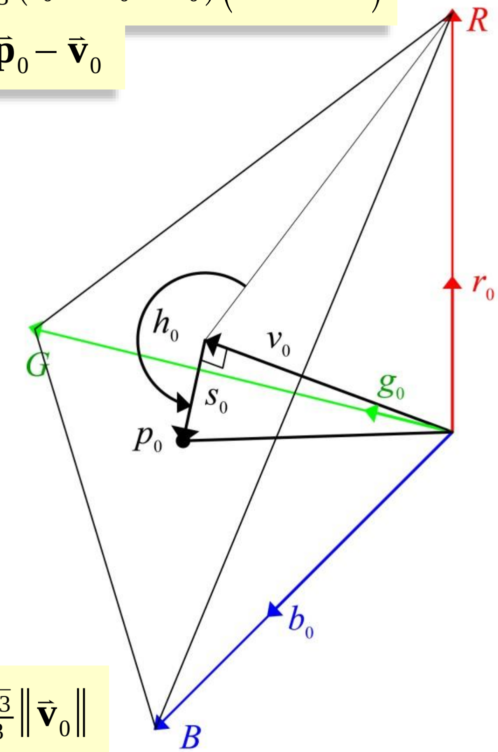


+ marks the spot where R=G=B= $(r_0+g_0+b_0)/3$



$$\vec{\mathbf{v}}_0 = \tfrac{1}{3}(r_0 + g_0 + b_0)(\hat{\mathbf{r}} + \hat{\mathbf{g}} + \hat{\mathbf{b}})$$

$$\vec{\mathbf{s}}_0 = \vec{\mathbf{p}}_0 - \vec{\mathbf{v}}_0$$



$p_0 = (r_0, g_0, b_0)$

Triangle:

lies in plane $R+G+B=(r_0+g_0+b_0)$,

intersects R, G, & B axes at $r_0+g_0+b_0$, and contains p_0.

$$v_0 = \tfrac{\sqrt{3}}{3}\left\|\vec{\mathbf{v}}_0\right\|$$

$$s_0 = \left\|\vec{\mathbf{s}}_0\right\|$$

$$h_0 = \angle\left\{\left[(r_0 + g_0 + b_0)\,\hat{\mathbf{r}} - \vec{\mathbf{v}}_0\right], \vec{\mathbf{s}}_0\right\}$$

RGB to HSV Conversion

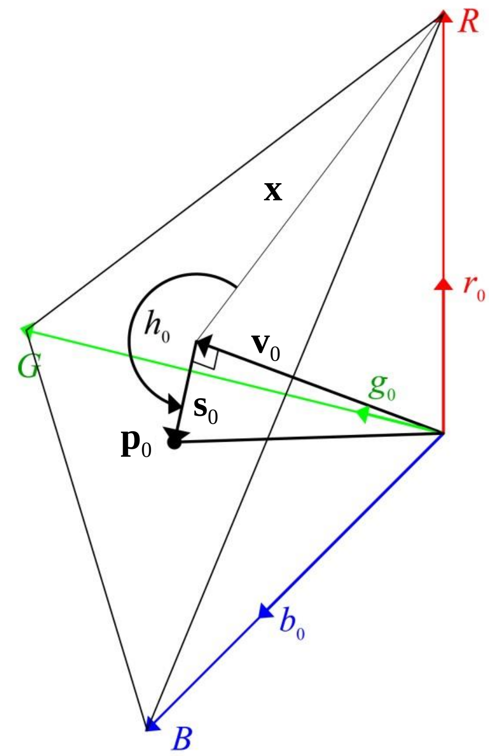
$$\mathbf{v}_0 = \tfrac{1}{3}\begin{bmatrix} c \\ c \\ c \end{bmatrix}, \quad \text{where } c = r_0 + g_0 + b_0.$$

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

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

v_0 is the value – a scalar.

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



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

RGB to HSV Conversion

$$\mathbf{v}_0 = \tfrac{1}{3}\begin{bmatrix} c \\ c \\ c \end{bmatrix}, \quad \text{where} \quad c = r_0 + g$$

c/3 is the usual value-image intensity (the average of *r*, *g*, & *b*), …

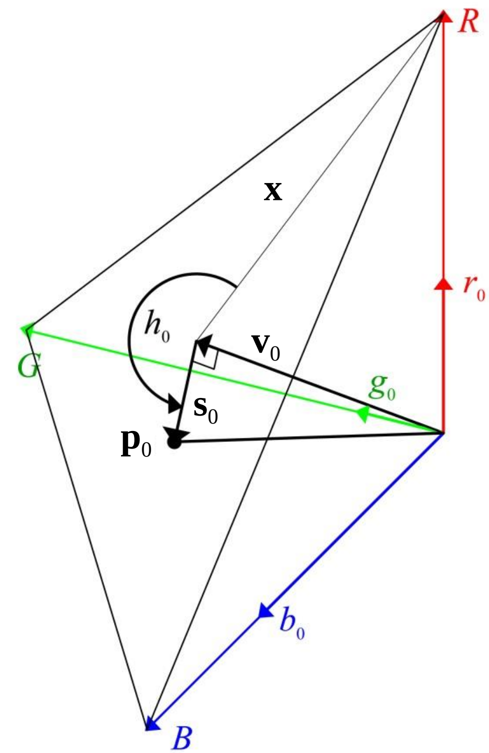
… so there is ambiguity in the definition of v_0. Its interpretation is contextual.

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

… c √3/3 is the length of the value vector,…

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

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



RGB to HSV Conversion

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

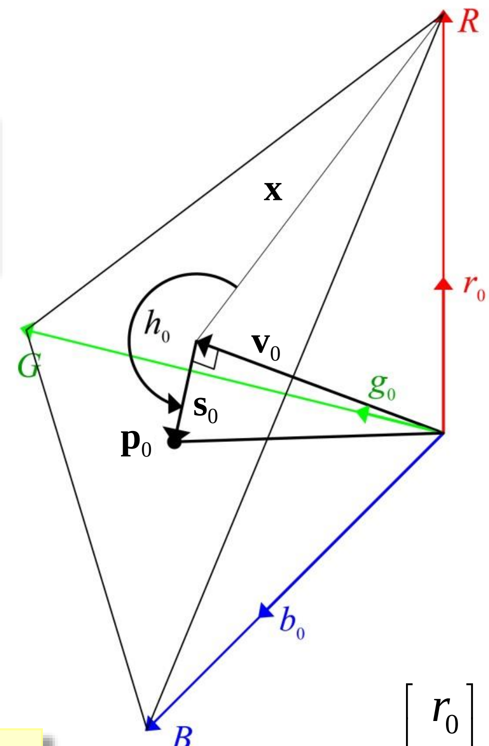
is the red line, $\mathbf{x}$, for **all** colors with value, $\mathbf{v}_0$.

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

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

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

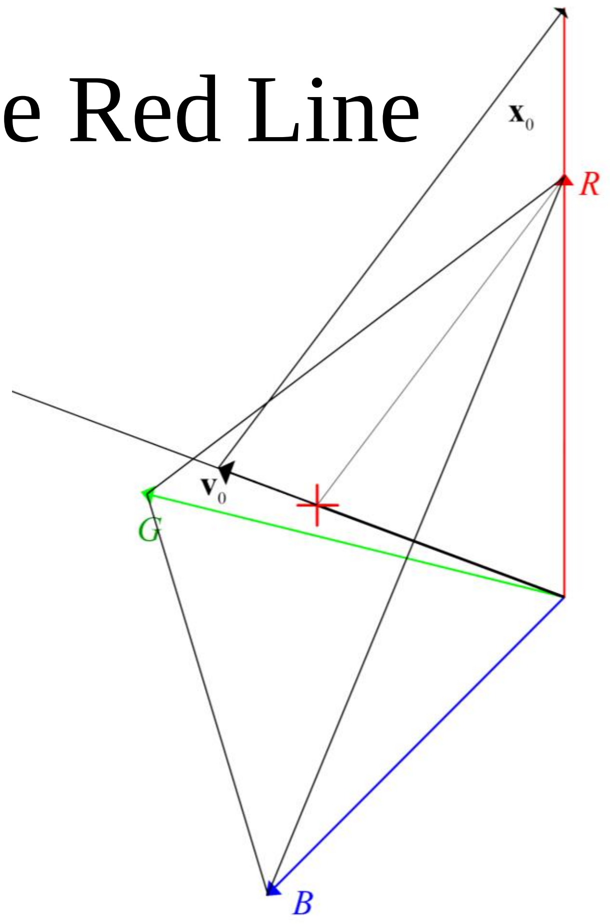
is the hue angle, h_0, for color $\mathbf{p}_0$.



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

RGB to HSV Conversion: The Red Line

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



Note that:

(1) for $c > 85$, the red line (vector) $\mathbf{x}$ extends beyond the color cube, and

(2) vector $\mathbf{x} = 0$ if and only if $c = 0$.

RGB to HSV Conversion

In summary,

$$v_0 = \tfrac{1}{3}c$$

The value is the average of the r, g, & b intensities of the color.

where $c = r_0 + g_0 + b_0$, the sum of the components of $\mathbf{p}_0$.

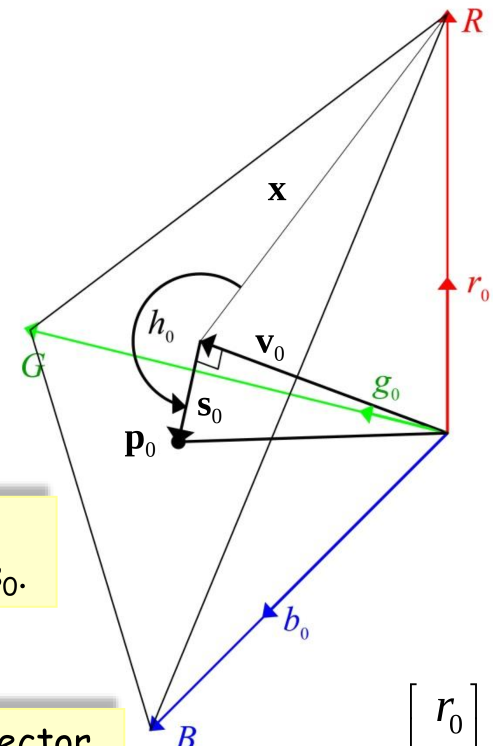
$$s_0 = \sqrt{(r_0 - v_0)^2 + (g_0 - v_0)^2 + (b_0 - v_0)^2},$$

The scalar saturation, s_0, is the length of the saturation vector, $\mathbf{s}_0$.

and

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

The dot product of (i) the unit vector in the saturation direction and (ii) the unit vector in the direction of the red line is the cosine of the hue.



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

Normalizing the Saturation

The scalar saturation,

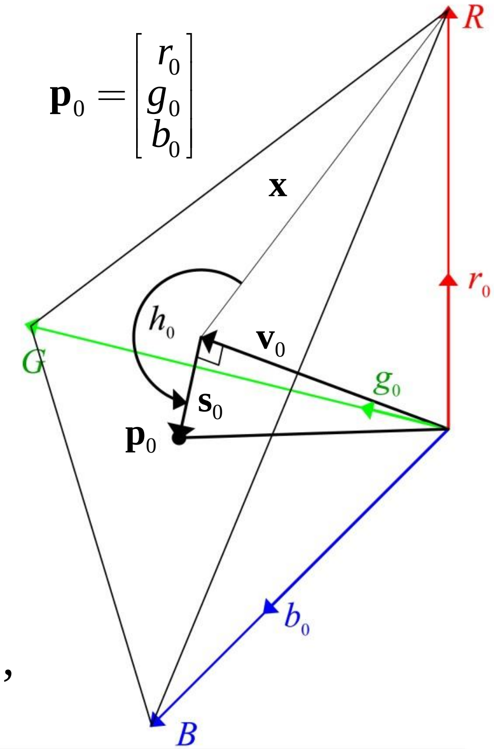
$$s_0 = \sqrt{\left(r_0 - v_0\right)^2 + \left(g_0 - v_0\right)^2 + \left(b_0 - v_0\right)^2},$$

usually is normalized to lie between 0 and 1. There are a number of possible ways to do this. One is to use the largest possible length of a saturation vector in the color cube. That vector lies in the triangle with vertices $[r\ \ g\ \ b]^{\mathrm{T}}$ = $[255\ \ 0\ \ 0]^{\mathrm{T}}$, $[0\ \ 255\ \ 0]^{\mathrm{T}}$, and $[0\ \ 0\ \ 255]^{\mathrm{T}}$. There are 3 such vectors, from the gray point to pure red, pure green, or pure blue. The red one is

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

which has length $s_{\max} = \left\| \mathbf{s}_{\max} \right\| \approx 208.2066.$

Therefore, s_0 can be replaced by $s_0 / s_{\max}$.



See other approaches to normalizing the saturation on slides nnn

Other Steps to Include in RGB→HSV

There are a few places in the algorithm where computation can be problematic. These include division by zero, exceeding limits due to round-off errors, and values returned by library functions that are inconsistent with the RGB→HSV algorithm.

Recall that the hue calculation for color, $\mathbf{p}$, requires division by the product of the length of the red vector, $\mathbf{x}(\mathbf{p})$, and the length of the saturation vector, $\mathbf{s}(\mathbf{p})$. If either of these is 0, then the argument could be undefined. Matlab returns `NaN` for `acos(0/0)`.

$$h = \theta = \cos^{-1}\left(\frac{\mathbf{s}_0}{\left\|\mathbf{s}_0\right\|} \bullet \frac{\mathbf{x}}{\left\|\mathbf{x}\right\|}\right).$$

If, due to finite word-length effects, the dot product evaluates to a number with magnitude ever-so-slightly greater than one, the inverse cosine will return a complex value.

$\mathbf{s}$ is the zero vector whenever $\mathbf{p}$ is a gray level (all its components are equal). Vector $\mathbf{x} = \mathbf{0}$ if and only if color $\mathbf{p} = \mathbf{0}$. The workaround is to add 1 to the denominator at every pixel where either $\mathbf{p} = 0$ or $s = 0$. Then set $h = 0$ at those same pixel locations. If h is complex, replace it with $\mathrm{real}(h)$.

Other Steps to Include in RGB→HSV

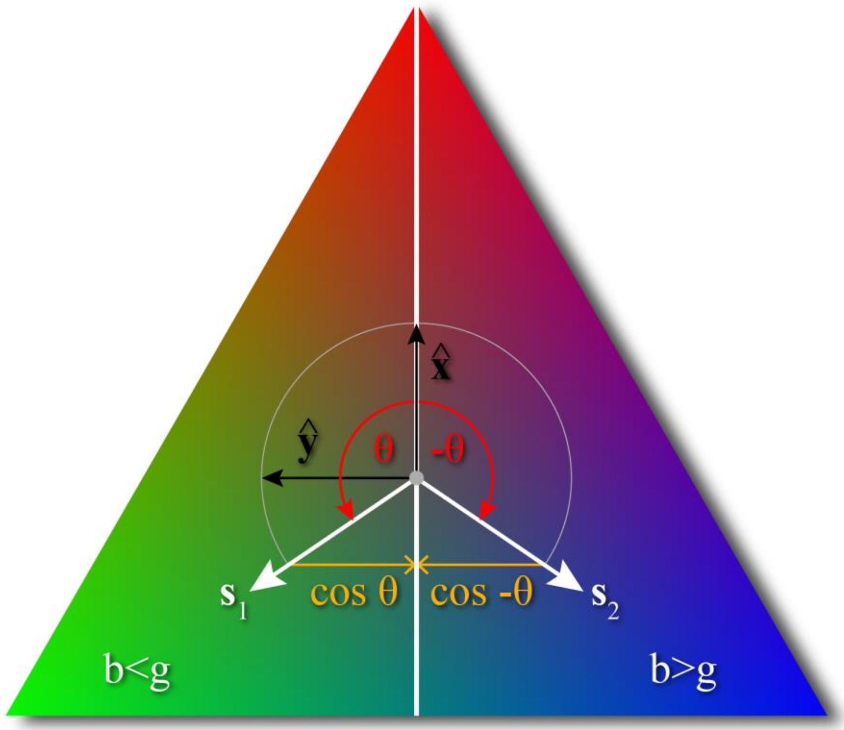
Inverse cosine routines, like Matlab's **`acos`**, **return angles in the range [0,π]**, whereas the RGB→HSV algorithm needs them to be in the range $[0,2\pi)$.

Note that $\theta \in [0,\pi]$ if and only if $b \leq g$ in $\mathbf{p}$ and $\theta \in (-\pi,0)$ if and only if $b > g$. Therefore the workaround is to let:

$$h = \begin{cases} \theta & \text{if} \quad b \leq g, \\ 2\pi - \theta & \text{if} \quad b > g, \end{cases}$$

where

$$\theta = \cos^{-1}\left(\frac{\mathbf{s}_0}{\|\mathbf{s}_0\|} \bullet \frac{\mathbf{x}}{\|\mathbf{x}\|} \right), \; \mathbf{p} = \begin{bmatrix} r \\ g \\ b \end{bmatrix}.$$



RGB to HSV Algorithm 1

A vector-geometric algorithm

1. Compute scalar value image, **v**, from RGB image, **I**=[**r g b**].
2. Compute vector value image, **V** = [**v v v**].
3. Compute vector saturation image, **S** = **I** − **V**.
4. Compute scalar saturation image, **s** = ‖**S**‖.
5. Compute red axis vector image, **X** = [2**v** −**v** −**v**].
6. Compute red axis scalar image, **x** = ‖**X**‖.
7. Create logical image, **z**(r,c) = (**s**(r,c)==0)|(**x**(r,c)==0).
8. Compute hue cosine image, **c** = (**S**•**X**)/((**s**·**x**)+**z**).
9. Compute hue angle image, **h** = $\cos^{-1}$(**c**).
10. Create logical image, **m**(r,c) = **b**(r,c) > **g**(r,c).
11. Adjust **h** = ~**m**·**h** + **m**·(2pi−**h**).
12. Normalize the saturation **s** = **s** / 208.2066.
13. Return $[\mathbf{h}\ \mathbf{s}\ \mathbf{v}]^T$.

||·|| is the norm operator – the square root of the pixel-wise sum of the squares of the image's 3-vector components.

RGB to HSV Algorithm 1

A vector-geometric algorithm

1. Compute scalar value image, **v**, from RGB image, **I**=[**r** **g** **b**].
2. Compute vector value image, **V** = [**v** **v** **v**].
3. Compute vector saturation image, **S** = **I** – **V**.
4. Compute scalar saturation image, **s** = ||**S**||.
5. Compute red axis vector image, **X** = [2**v** –**v** –**v**
6. Compute red axis scalar image, **x** = ||**X**||.
7. Create logical image, **z**(r,c) = (**s**(r,c)==0)|(
8. Compute hue cosine image, **c** = (**S**•**X**)/((**s**·**x**)+**z**)
9. Compute hue angle image, **h** = $\cos^{-1}$(**c**).
10. Create logical image, **m**(r,c) = **b**(r,c) > **g**(r,c).
11. Adjust **h** = ~**m**·**h** + **m**·(2pi–**h**).
12. Normalize the saturation **s** = **s** / 208
13. Return [**h** **s** **v**]^T.

V =3-band val img

S•**X** is the pixel-wise dot product of images **S** & **X**. **s**·**x** is the p.w. scalar product of the scalar images **s** and **x**.

z and **m** are binary images
z(r,c)==1 iff **s**(r,c)==0;
m(r,c)==1 iff **b**(r,c) > **g**(r,c).

matlab: ||**S**|| = sqrt(sum(**S**.^2,3));

A Faster Hue Calculation

Project color $\mathbf{p}_0$ onto the red line, $\mathbf{x}$, using the dot product,

$$\mathbf{x} \bullet \mathbf{p}_0 = \frac{c}{3}\begin{bmatrix} 2 \\ -1 \\ -1 \end{bmatrix} \bullet \begin{bmatrix} r_0 \\ g_0 \\ b_0 \end{bmatrix} = \frac{c}{3}\left[2r_0 - g_0 - b_0\right].$$

Vector $\mathbf{z} = \sqrt{3}\,\tfrac{c}{3}\begin{bmatrix} 0 & 1 & -1 \end{bmatrix}^{\mathrm{T}}$ is perpendicular to $\mathbf{x}$. Project $\mathbf{p}_0$ on to $\mathbf{z}$ to get

$$\mathbf{z} \bullet \mathbf{p}_0 = \frac{c\sqrt{3}}{3}\begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} \bullet \begin{bmatrix} r_0 \\ g_0 \\ b_0 \end{bmatrix} = \frac{c\sqrt{3}}{3}\left[g_0 - b_0\right].$$

The ratio of the second dot product to the first is the tangent of the hue angle. Therefore,

$$h = \tan^{-1}\left(\frac{\mathbf{z} \bullet \mathbf{p}_0}{\mathbf{x} \bullet \mathbf{p}_0}\right) = \tan^{-1}\left(\frac{\sqrt{3}\left(g_0 - b_0\right)}{2r_0 - g_0 - b_0}\right).$$

RGB to HSV Algorithm 2

A vector-geometric algorithm

Using the expressions on the previous slide reduces the number of variables and the number of computations necessary to compute the hue. Experiments with Matlab show Algorithm 2 to be 2.7 times faster than Algorithm 1. Both algorithms return the same values.

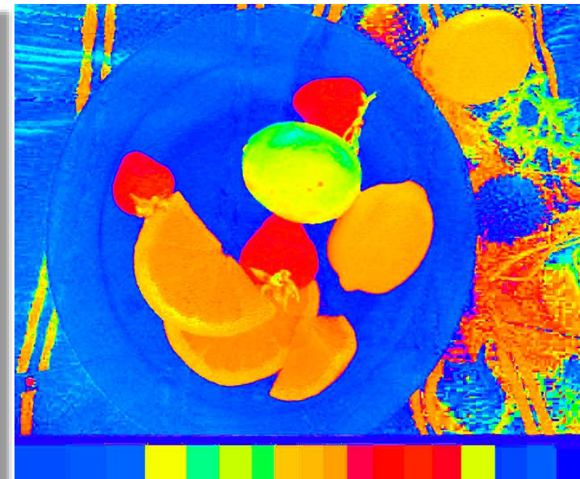
1. Compute scalar value image, **v**, from RGB image, $\mathbf{I}=[\mathbf{r}\ \mathbf{g}\ \mathbf{b}]^T$.
2. Compute vector value image, $\mathbf{V} = [\mathbf{v}\ \mathbf{v}\ \mathbf{v}]^T$.
3. Compute vector saturation image, $\mathbf{S} = \mathbf{I} - \mathbf{V}$.
4. Compute scalar saturation image, $\mathbf{s} = \|\mathbf{S}\|$.
5. Compute the scalar image, $\mathbf{x} = 2\mathbf{r}-\mathbf{g}-\mathbf{b}$.
6. Compute the scalar image, $\mathbf{y} = \mathrm{sqrt}(3)(\mathbf{g}-\mathbf{b})$.
7. Compute hue angle image, $\mathbf{h} = \mathrm{atan2}(\mathbf{y},\mathbf{x})$.
8. Create a logical image, $\mathbf{m}(r,c) = \mathbf{b}(r,c) > \mathbf{g}(r,c)$.
9. Adjust $\mathbf{h} = \sim\mathbf{m}\cdot\mathbf{h} + \mathbf{m}\cdot(2pi+\mathbf{h})$.
10. Normalize $\mathbf{s} = \mathbf{s} / 208.2066$.
11. Return $[\mathbf{h}\ \mathbf{s}\ \mathbf{v}]^T$.

atan2 returns angles in $(-\pi,\pi)$. Negative angles are in the b>g region. They are shifted by adding them to 2π.

Example RGB Image and HSV Bands

Original image Karen Gillis Taylor, “A Fruit Color Study.”

Hue image displayed with a primary color color-map



Value Image

Saturation image scaled to (0,…,255)

HSV to RGB Conversion

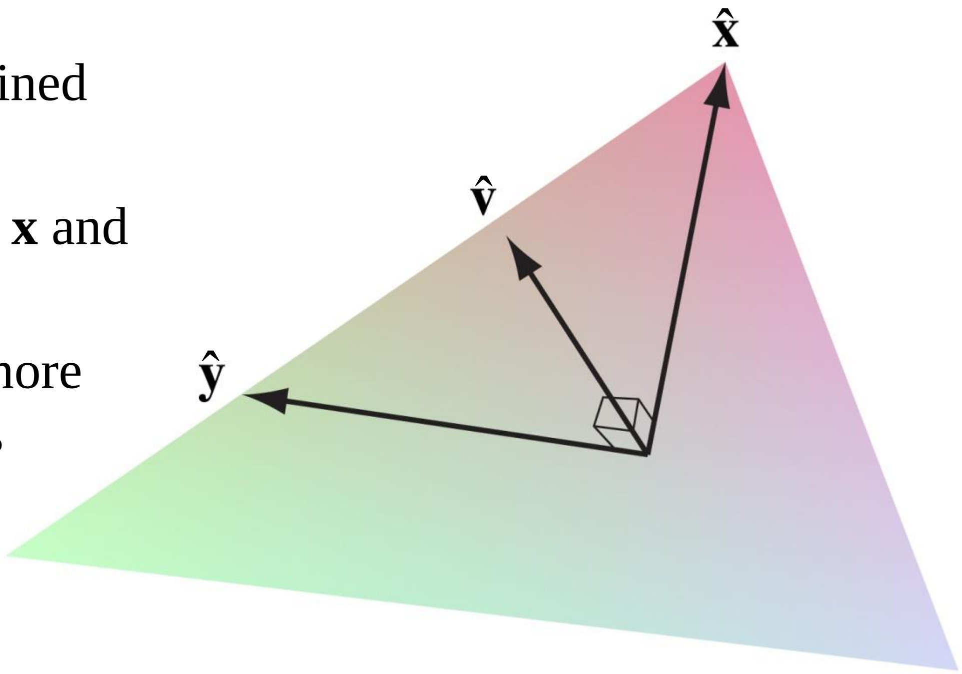
This conversion requires a change of coordinates through a rotation and a translation.

The equivalue plane is perpendicular to the value vector, **v**.

The plane contains vector **x** defined on slide 77.

Therefore, **v** is perpendicular to **x** and **y** = **v** × **x** is also in the plane.

If we keep the directions but ignore the magnitudes, the unit vectors



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

form an orthonormal basis with respect to the equivalue plane.

HSV to RGB Conversion

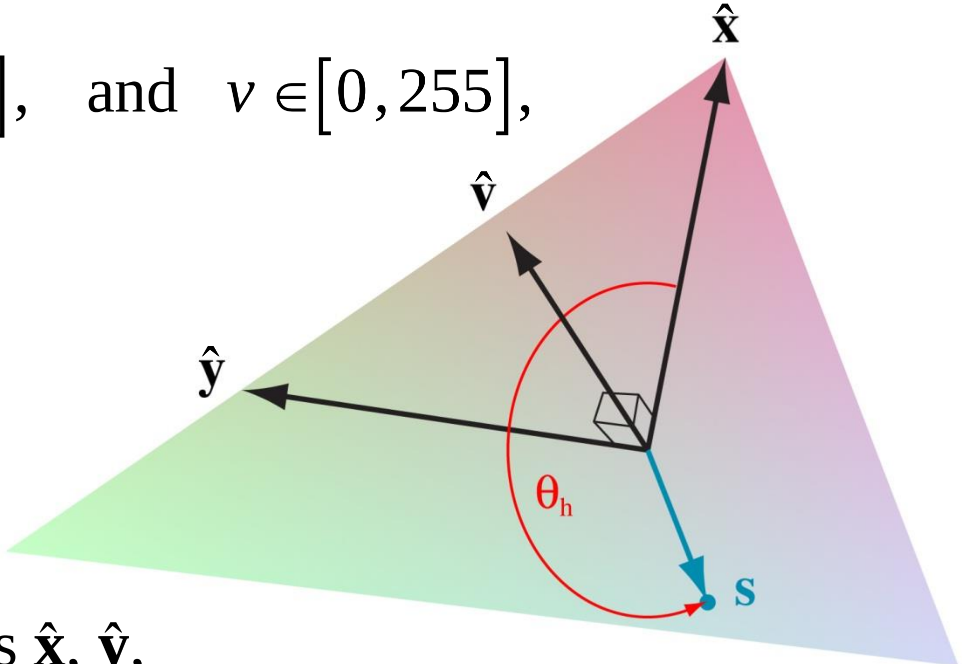
Since the x- and y-axes lie in the equivalue plane and the cdt. origin is the gray point, we set v = 0 for now.

Given values h, s, and v, where

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

the saturation vector is

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



with respect to unit vectors $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{v}}$, in the equivalue plane.

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

HSV to RGB Conversion

If s is in the range 0 to 1, then it must be denormalized first by multiplying by smax.

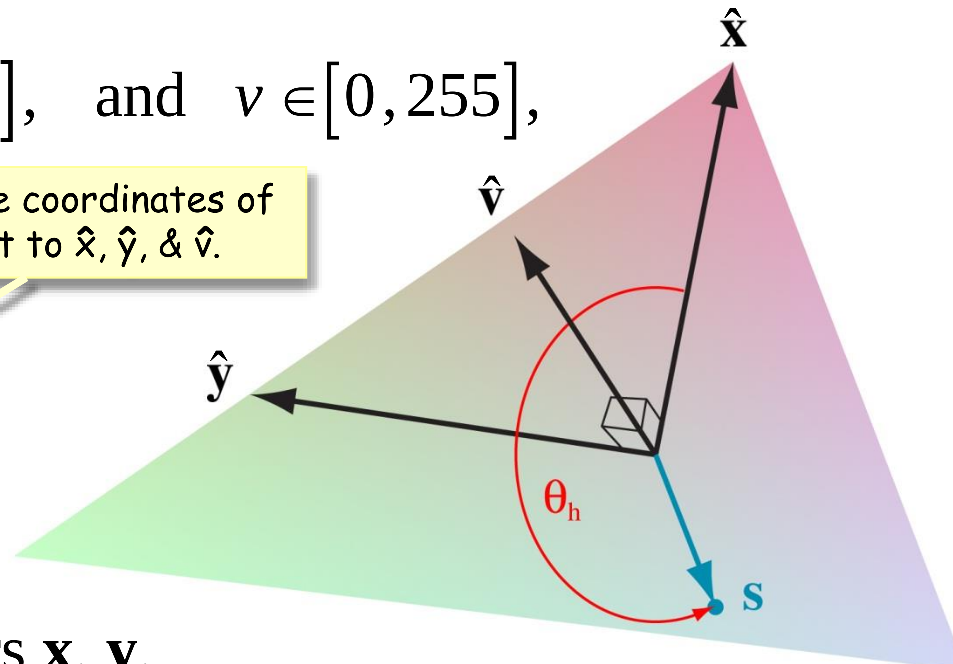
Given values h, s, and v, where

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

the saturation ve

These are the coordinates of **s** with respect to $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, & $\hat{\mathbf{v}}$.

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



with respect to unit vectors **x**, **y**, and

This is **s** written as a linear combination of vectors $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, & $\hat{\mathbf{v}}$.

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

HSV to RGB Conversion

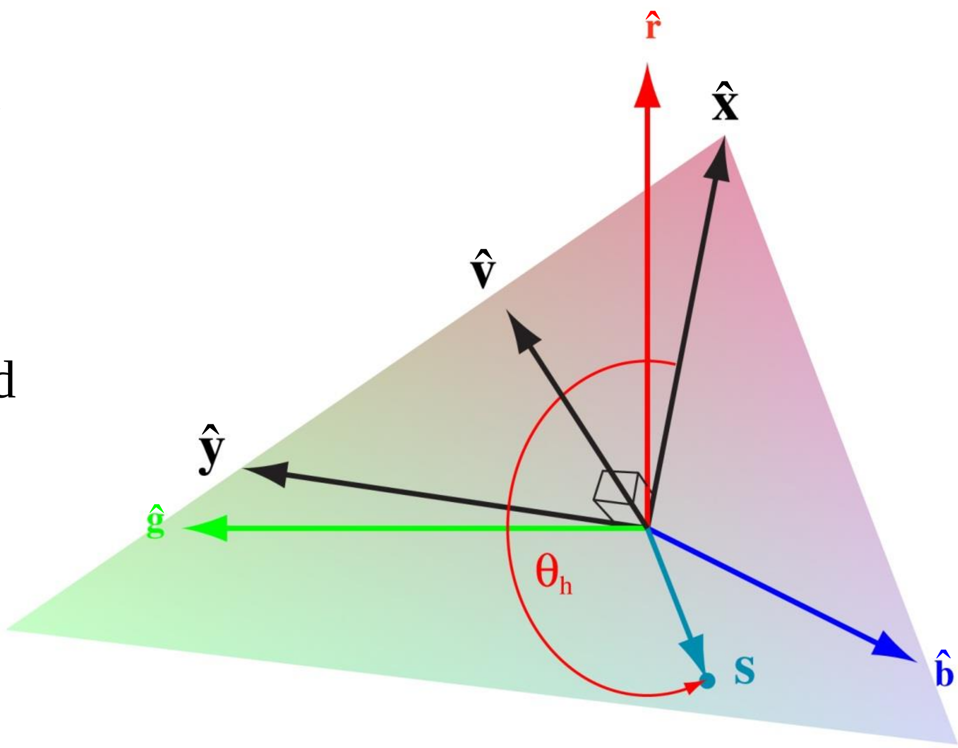
$\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{v}}$, are not in the same directions as the red, green, and blue unit vectors, $\hat{\mathbf{r}}$, $\hat{\mathbf{g}}$, and $\hat{\mathbf{b}}$.

Therefore, $[\mathbf{s}]_{\mathbf{xyv}}$, which we know, is not equal to $[\mathbf{s}]_{\mathbf{rgb}}$, which we do not know, but need in order to find the color, $\mathbf{p}_0$, with respect to $\hat{\mathbf{r}}$, $\hat{\mathbf{g}}$, and $\hat{\mathbf{b}}$.

$$[\mathbf{s}]_{\mathbf{rgb}} = \begin{bmatrix} r_0 & g_0 & b_0 \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{s} \leftrightarrow r_0 \hat{\mathbf{r}} + g_0 \hat{\mathbf{g}} + b_0 \hat{\mathbf{b}},$$

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



We need to find r_0, g_0, & b_0...

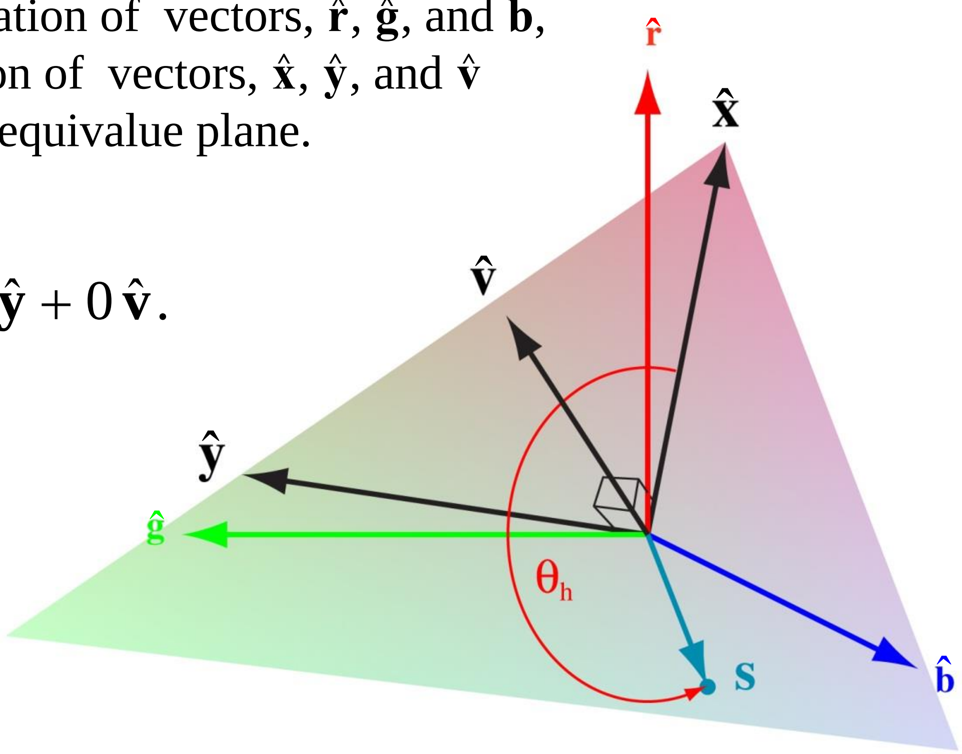
HSV to RGB Conversion

… which means we need to find the x, y, & v unit vectors in terms of the r, g, & b unit vectors.

Vector **s** written as a linear combination of vectors, $\hat{\mathbf{r}}$, $\hat{\mathbf{g}}$, and $\hat{\mathbf{b}}$, and **s** written as a linear combination of vectors, $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{v}}$ both refer to the same point on the equivalue plane.

$$\mathbf{s} \leftrightarrow r_0\hat{\mathbf{r}} + g_0\hat{\mathbf{g}} + b_0\hat{\mathbf{b}},$$

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



The specific numbers in $[\mathbf{s}]_{\mathbf{rgb}}$ and in $[\mathbf{s}]_{\mathbf{xyv}}$ (that represent the point w.r.t. the two coordinate systems) are, however, different.

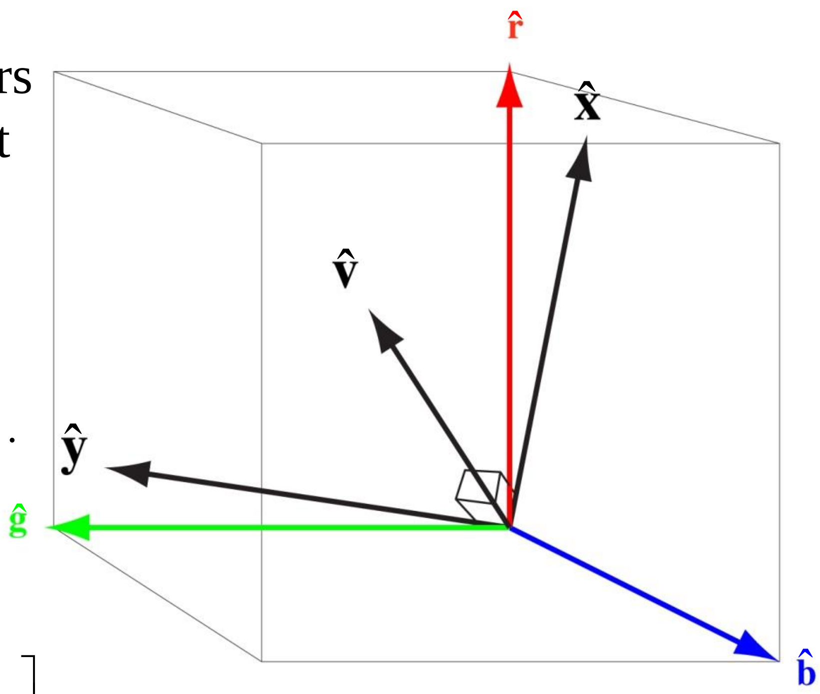
$$[\mathbf{s}]_{\mathbf{rgb}} = \begin{bmatrix} r_0 & g_0 & b_0 \end{bmatrix}^{\mathrm{T}} \text{ and}$$

$$[\mathbf{s}]_{\mathbf{xyz}} = \begin{bmatrix} s\cos(h) & s\sin(h) & 0 \end{bmatrix}^{\mathrm{T}} \text{ but } [\mathbf{s}]_{\mathbf{rgb}} \neq [\mathbf{s}]_{\mathbf{xyz}}$$

HSV to RGB Conversion

We can find r_0, g_0, and b_0, from h_0, s_0, and v_0, if we know how the unit vectors $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{v}}$, are expressed with respect to $\hat{\mathbf{r}}$, $\hat{\mathbf{g}}$, and $\hat{\mathbf{b}}$. That relationship is in the form of a rotation matrix, A, such that,

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



Then,

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

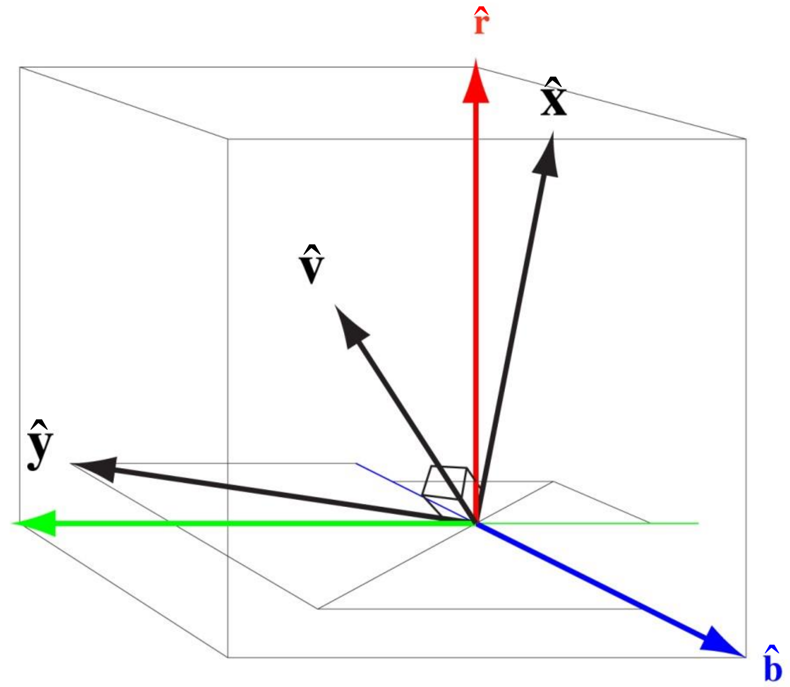
HSV to RGB Conversion

When written w.r.t the **xyz** coordinate system we have

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

So that,

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



But that implies,

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

The columns of the rotation matrix, A, are the x, y, & v unit vectors in rgb coordinates.

HSV to RGB Conversion

How to find the x, y, & z unit vectors in rgb coordinates:

$\hat{\mathbf{v}}$ is the unit vector in the direction $[1\ 1\ 1]^T$ when written w.r.t **rgb** coordinates.

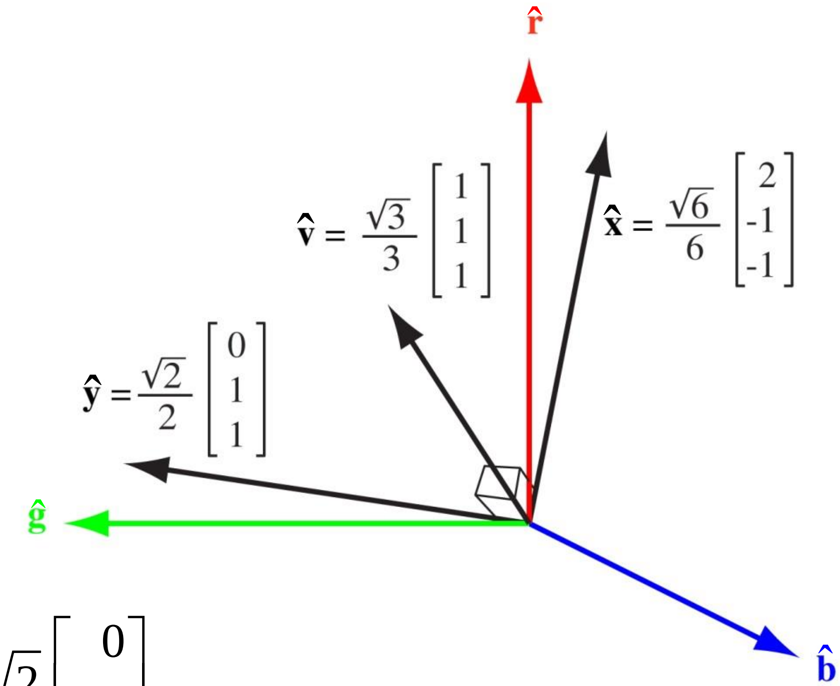
$$\left[\hat{\mathbf{v}}\right]_{\mathbf{rgb}} = \frac{\sqrt{3}}{3}\begin{bmatrix}1\\1\\1\end{bmatrix}$$

$\hat{\mathbf{x}}$ is perpendicular to $\hat{\mathbf{v}}$ and has equal $\hat{\mathbf{g}}$ and $\hat{\mathbf{b}}$ components.

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

$\hat{\mathbf{y}}$ is the cross product of $\hat{\mathbf{v}}$ with $\hat{\mathbf{x}}$.

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



HSV to RGB Conversion

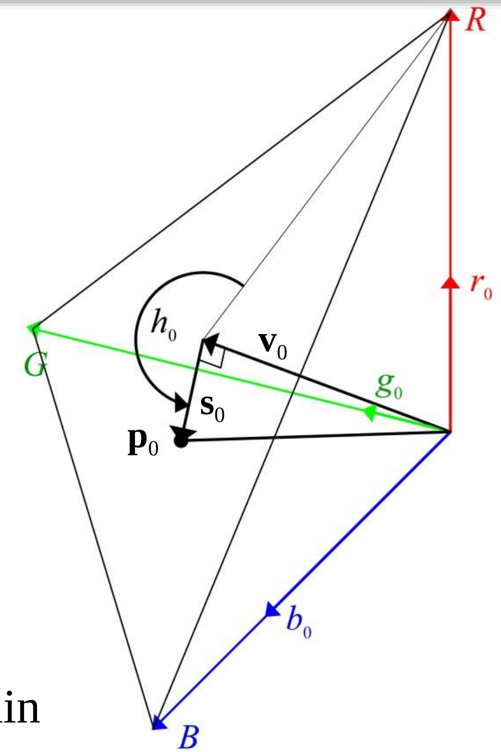
The x, y, & z unit vectors in rgb coordinates are the columns of the rotation matrix:

Therefore, the rotation matrix is

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

Substitute that into the 2nd equation on slide 93 to get:

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



Finally, $[\mathbf{s}]_{\mathbf{rgb}}$ must be translated to the value vector to obtain the rgb color of $\mathbf{p}_0$:

$$\mathbf{p}_0 = [\mathbf{p}]_{\mathbf{rgb}} = [\mathbf{s}]_{\mathbf{rgb}} + [\mathbf{v}]_{\mathbf{rgb}}$$, where $\mathbf{s}_0 = [\mathbf{s}]_{\mathbf{rgb}}$ and $[\mathbf{v}]_{\mathbf{rgb}} = \mathbf{v}_0$ as def'd. on slide 75.

HSV to RGB Algorithm

A vector-geometric algorithm

This inverts the outputs of both Algorithm 1 and Algorithm 2.

1. Reshape R×C images h, s, and v into RC×1 column vectors.
2. Denormalize the saturation image, $\mathbf{s} = 208.2066 \cdot \mathbf{s}$.
3. Compute the x-coordinate image, $\mathbf{x} = \mathbf{s} \cdot \cos(\mathbf{h})$.
4. Compute the y-coordinate image, $\mathbf{y} = \mathbf{s} \cdot \sin(\mathbf{h})$.
5. Construct the value vector image, $\mathbf{V} = [\mathbf{v}\ \mathbf{v}\ \mathbf{v}]^T$
6. Construct the 1st two columns of rotation matrix A
7. Rotate the saturation image $\mathbf{S} = [\mathbf{s}_1\ \mathbf{s}_2\ \mathbf{s}_3]^T = A[\mathbf{x}\ \mathbf{y}]^T$.
8. Add the result to $\mathbf{V}$ to recover $\mathbf{I}_{RGB} = \mathbf{S} + \mathbf{V}$.
9. Round $\mathbf{I}_{RGB}$, convert it to uint8, and reshape it to R×C×3.

Note: if size(X) == [R C B], then size(X(:)) == [R·C·B 1].

Note: V is 3xRC, since v is an RCx1 column vector and $V = [v\ v\ v]^T$.
A_{3x2} multiplies $[x\ y]^T$ which is 2xRC.
Lines 7 & 8: S, V, and I_{RGB} are 3xRC.

Alternative HSV Representations

The HSV algorithms presented in the first part of this lecture are direct implementations of the vector math.

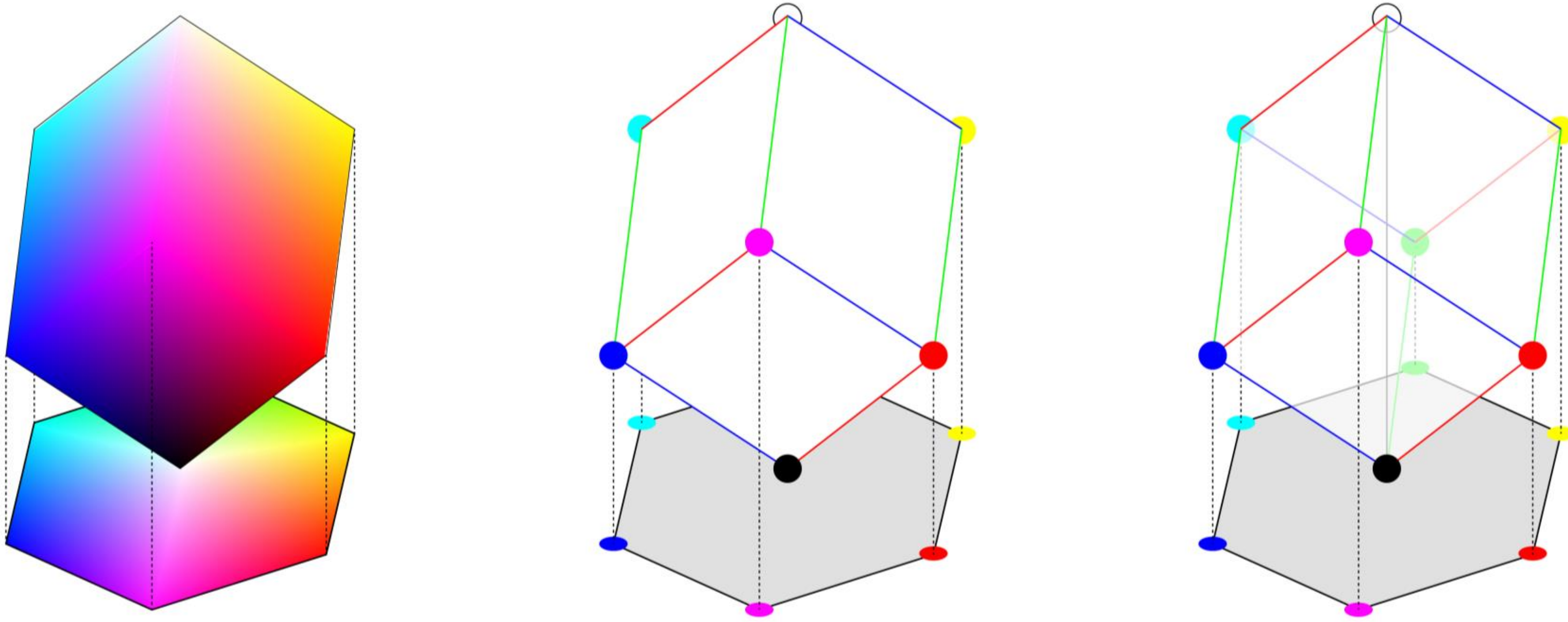
One of the problems with the direct approach is that it is that the presence of transcendental functions increases the time necessary to convert an image.

To use HSV at video frame rate requires a simpler algorithm that yields a reasonable approximation to the direct implementation.

```
1.  v=(r+g+b)/3.
2.  V = [v v v]^T.
3.  S = I - V.
4.  s = ||S||.
5.  X = [2v -v -v]^T.
6.  x = ||X||.
7.  z(r,c) = …
     (s(r,c)==0)|(x(r,c)==0).
8.  c = (S•X)/((s·x)+z).
9.  h = cos^-1(c).
10. m(r,c) = b(r,c) > g(r,c).
11. h = ~m·h + m·(2pi-h).
12. s = s / 208.2066.
13.  Return [h s v]^T.
```

Hexagonal Projection Representations

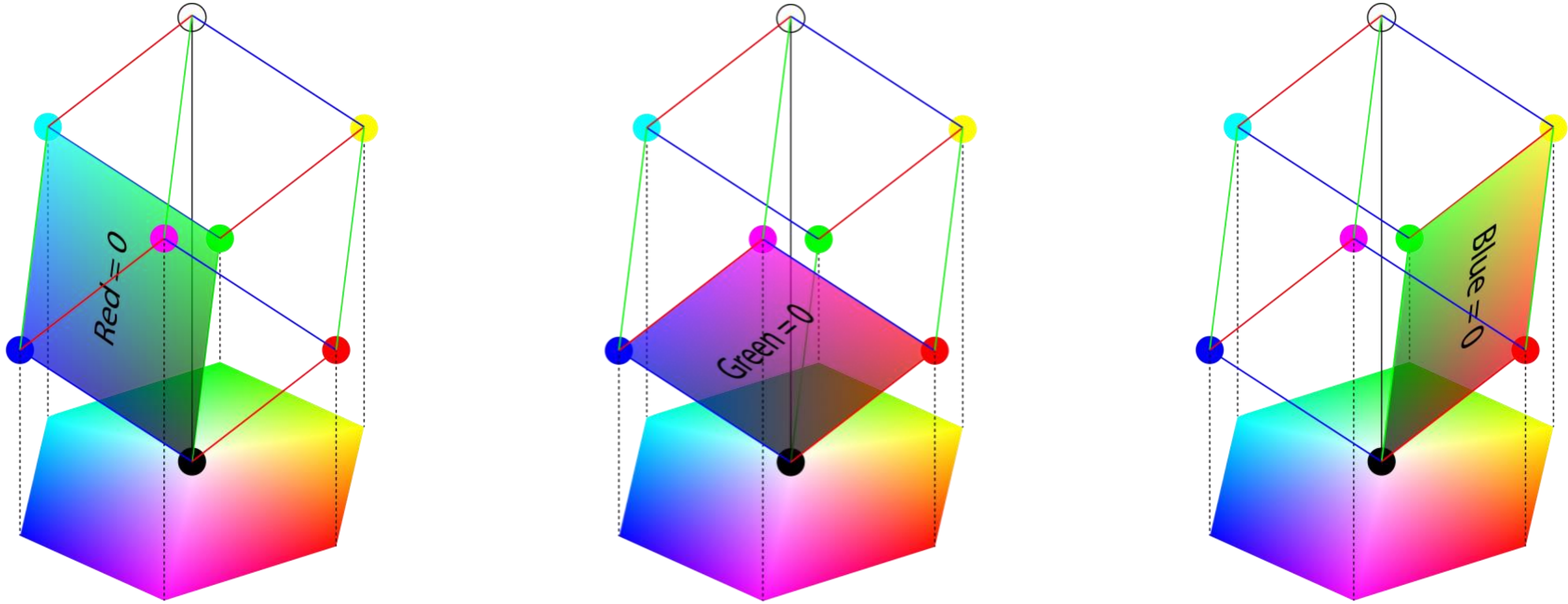
Faster algorithms use a hexagonal projection of the color cube. To generate the hexagon, rotate the color cube in 3-space then project it along its gray line (the diagonal from white to black) onto a chromaticity plane. The result is a hexagon that has the 6 primary colors – red, yellow, green, cyan, blue, and magenta – at its vertices.



rgb color vector, **p**, is projected on the hexagon where its hue and saturation are computed.

Hexagonal Projection Representations

Faster algorithms use a hexagonal projection of the color cube. To generate the hexagon, rotate the color cube in 3-space then project it along its gray line (the diagonal from white to black) onto a chromaticity plane. The result is a hexagon that has the 6 primary colors – red, yellow, green, cyan, blue, and magenta – at its vertices.



rgb color vector, **p**, is projected on the hexagon where its hue and saturation are computed.

A Fast RGB to HSV Algorithm

Given color $\mathbf{p}_0 = [R\ G\ B]^T$ where $R, G, B \in \{0,\ldots,255\}$, to compute $[h\ s\ v]^T$ where s, v $\in[0,1]$ and h $\in[0,360)$ the algorithm proceeds as follows:

```
1. Compute [r g b] = [R G B]/255.
2. Set m = min(r,g,b) , M = max(r,g,b).
3. Set v = M.
4. Compute C = M – m.
5. If C == 0 then s=0, h=0. Return [h s v]^T.
6. s = C/M.
7. If M==r then h = ((g-b)/c) modulo 6.
8. else if M==g then h = 2 + (b-r)/c.
9. else h = 4 + (r-g)/c.
10.h = 60h.
```

R,G,B are numbers here not images.

Experiments with Matlab show this algorithm to be 3 times faster than Algorithm 1 and 1.13 faster than Algorithm 2. The numbers output by this one differ from the other two.

Reference: HSL and HSV - Wikipedia, the free encyclopedia

A Fast HSV to RGB Algorithm

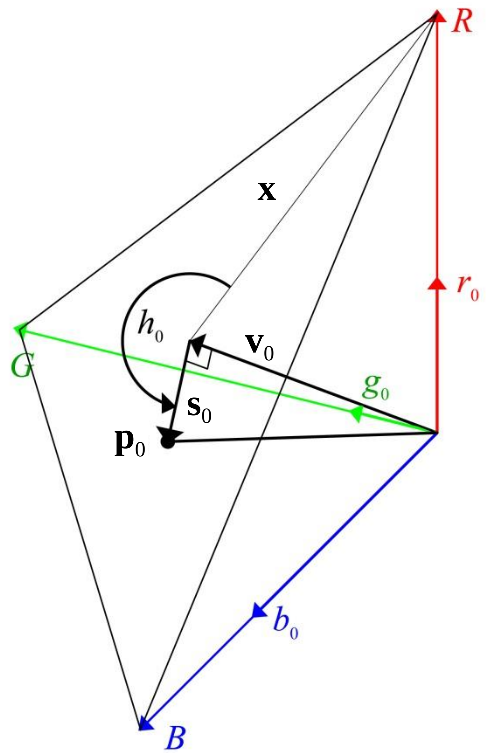
Given vector $\mathbf{h}_0^T = [h\ s\ v]$ where $h \in [0, 360)$, $s \in [0, 1]$, and $v \in [0, 1]$, to compute $\mathbf{p}_0^T = [r\ g\ b]$ where $r, g, b \in \{0, \ldots, 255\}$:

```
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]
```

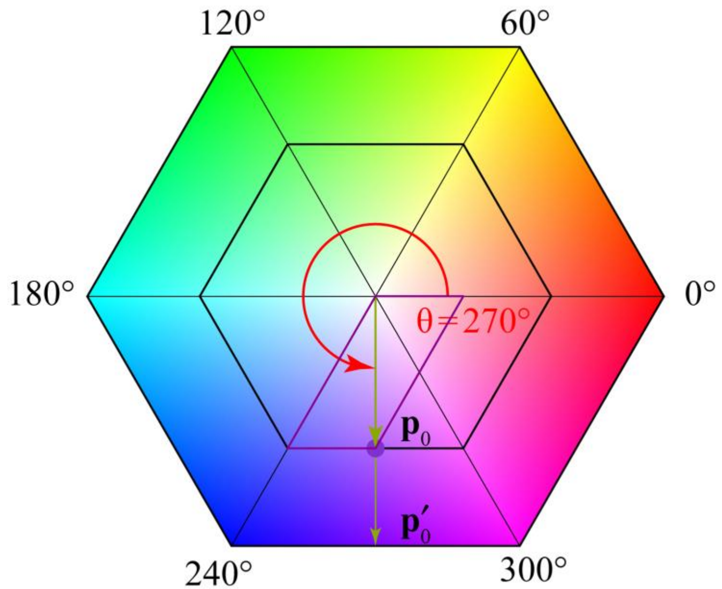
h, s, & v are numbers here not images.

Reference: HSL and HSV - Wikipedia, the free encyclopedia

Comparison of the Two Algorithms



Geometric representation
on the equivalue plane



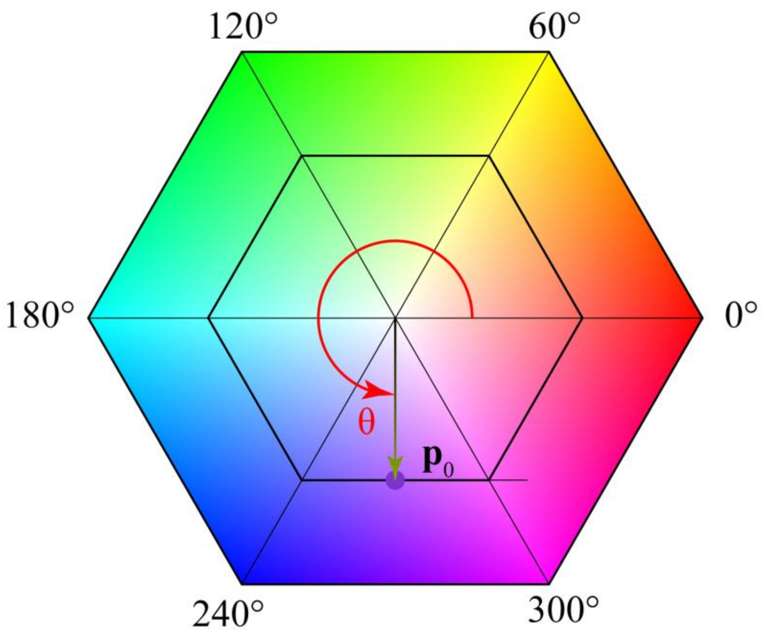
Projection of the color on
the choma-hexagon.

Geometric Hexagonal Projection

Given a color, its **value** is the projection, parallel to the hexagon, of the rgb point onto the gray line scaled by $\sqrt{3}/3$.

Its **hue** is the angular distance around the hexagon from the projection of the red axis to the color, projected parallel to the gray line.

The **saturation** is the ratio of the distance from the gray line to the projected color, $[0,\mathbf{p}_0]$, to the distance from the gray line to one of the vertices of the hexagon.



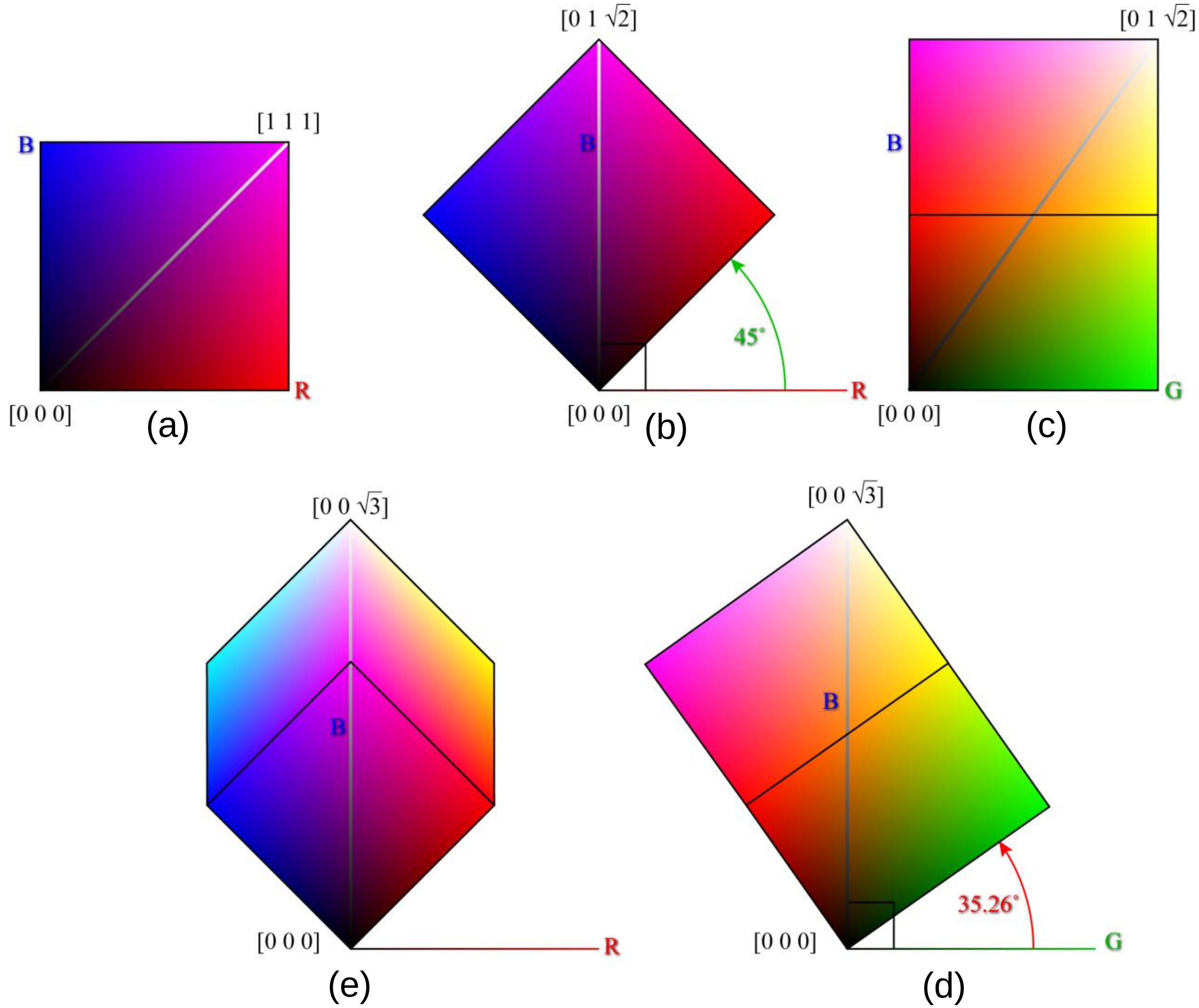
To compute the projection of a color we must find the rotation matrix for the cube.

Reference: HSL and HSV - Wikipedia, the free encyclopedia

Rotation of the Color Cube

Rotations for projection onto the hexagon:

(a) Original cube viewed along the green axis.
(b) Rotated 45° around the green axis.
(c) Rotated cube viewed along the red axis.
(d) Rotated 35.26° around the red axis.
(e) Final pose viewed along the green axis.



Rotation of the Color Cube

Rotation matrix:

$$R_{\text{hex}} = R_{\text{red}} R_{\text{green}} = \frac{1}{6}\begin{bmatrix} 3\sqrt{2} & 0 & -3\sqrt{2} \\ -\sqrt{6} & 2\sqrt{6} & -\sqrt{6} \\ 2\sqrt{3} & 2\sqrt{3} & 2\sqrt{3} \end{bmatrix}, \text{ where } R_{\text{green}} = \frac{1}{2}\begin{bmatrix} \sqrt{2} & 0 & -\sqrt{2} \\ 0 & 2 & 0 \\ \sqrt{2} & 0 & \sqrt{2} \end{bmatrix}, \ \& \ R_{\text{red}} = \frac{1}{3}\begin{bmatrix} 3 & 0 & 0 \\ 0 & \sqrt{6} & -\sqrt{3} \\ 0 & \sqrt{3} & \sqrt{6} \end{bmatrix}.$$

Cube corners:

$$\Sigma_{\text{CC}} = \{K,R,Y,G,C,B,M,W\} = 255\cdot\left\{\begin{bmatrix}0\\0\\0\end{bmatrix},\begin{bmatrix}1\\0\\0\end{bmatrix},\begin{bmatrix}1\\1\\0\end{bmatrix},\begin{bmatrix}0\\1\\0\end{bmatrix},\begin{bmatrix}0\\1\\1\end{bmatrix},\begin{bmatrix}0\\0\\1\end{bmatrix},\begin{bmatrix}1\\0\\1\end{bmatrix},\begin{bmatrix}1\\1\\1\end{bmatrix}\right\}$$

Rotated cube corners:

$$R\Sigma_{\text{CC}} = \frac{255}{6}\left\{\begin{bmatrix}0\\0\\0\end{bmatrix},\begin{bmatrix}3\sqrt{2}\\-\sqrt{6}\\2\sqrt{3}\end{bmatrix},\begin{bmatrix}3\sqrt{2}\\\sqrt{6}\\4\sqrt{3}\end{bmatrix},\begin{bmatrix}0\\2\sqrt{6}\\2\sqrt{3}\end{bmatrix},\begin{bmatrix}-3\sqrt{2}\\\sqrt{6}\\4\sqrt{3}\end{bmatrix},\begin{bmatrix}-3\sqrt{2}\\-\sqrt{6}\\2\sqrt{3}\end{bmatrix},\begin{bmatrix}0\\-2\sqrt{6}\\4\sqrt{3}\end{bmatrix},\begin{bmatrix}0\\0\\6\sqrt{3}\end{bmatrix}\right\}$$

Projected, rotated cube corners:

$$\Phi\{R\Sigma_{\text{CC}}\} = \frac{255}{6}\left\{\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}3\sqrt{2}\\-\sqrt{6}\end{bmatrix},\begin{bmatrix}3\sqrt{2}\\\sqrt{6}\end{bmatrix},\begin{bmatrix}0\\2\sqrt{6}\end{bmatrix},\begin{bmatrix}-3\sqrt{2}\\\sqrt{6}\end{bmatrix},\begin{bmatrix}-3\sqrt{2}\\-\sqrt{6}\end{bmatrix},\begin{bmatrix}0\\-2\sqrt{6}\end{bmatrix},\begin{bmatrix}0\\0\end{bmatrix},\right\}$$

Here's the math.

Rotation of the Color Cube

Rotation matrix:

45° around green

35.26° around red

$$R_{\text{hex}} = R_{\text{red}} R_{\text{green}} = \frac{1}{6}\begin{bmatrix} 3\sqrt{2} & 0 & -3\sqrt{2} \\ -\sqrt{6} & 2\sqrt{6} & -\sqrt{6} \\ 2\sqrt{3} & 2\sqrt{3} & 2\sqrt{3} \end{bmatrix}, \text{ where } R_{\text{green}} = \frac{1}{2}\begin{bmatrix} \sqrt{2} & 0 & -\sqrt{2} \\ 0 & 2 & 0 \\ \sqrt{2} & 0 & \sqrt{2} \end{bmatrix}, \ \& \ R_{\text{red}} = \frac{1}{3}\begin{bmatrix} 3 & 0 & 0 \\ 0 & \sqrt{6} & -\sqrt{3} \\ 0 & \sqrt{3} & \sqrt{6} \end{bmatrix}.$$

Cube corners:

$$\Sigma_{CC} = \{K,R,Y,G,C,B,M,W\} = 255\cdot\left\{\begin{bmatrix}0\\0\\0\end{bmatrix},\begin{bmatrix}1\\0\\0\end{bmatrix},\begin{bmatrix}1\\1\\0\end{bmatrix},\begin{bmatrix}0\\1\\0\end{bmatrix},\begin{bmatrix}0\\1\\1\end{bmatrix},\begin{bmatrix}0\\0\\1\end{bmatrix},\begin{bmatrix}1\\0\\1\end{bmatrix},\begin{bmatrix}1\\1\\1\end{bmatrix}\right\}$$

6 primaries + black and white

Rotated cube corners:

$$R\Sigma_{CC} = \frac{255}{6}\left\{\begin{bmatrix}0\\0\\0\end{bmatrix},\begin{bmatrix}3\sqrt{2}\\-\sqrt{6}\\2\sqrt{3}\end{bmatrix},\begin{bmatrix}3\sqrt{2}\\\sqrt{6}\\4\sqrt{3}\end{bmatrix},\begin{bmatrix}0\\2\sqrt{6}\\2\sqrt{3}\end{bmatrix},\begin{bmatrix}-3\sqrt{2}\\\sqrt{6}\\4\sqrt{3}\end{bmatrix},\begin{bmatrix}-3\sqrt{2}\\-\sqrt{6}\\2\sqrt{3}\end{bmatrix},\begin{bmatrix}0\\-2\sqrt{6}\\4\sqrt{3}\end{bmatrix},\begin{bmatrix}0\\0\\6\sqrt{3}\end{bmatrix}\right\}$$

Now the 3rd dimension is along the gray line.

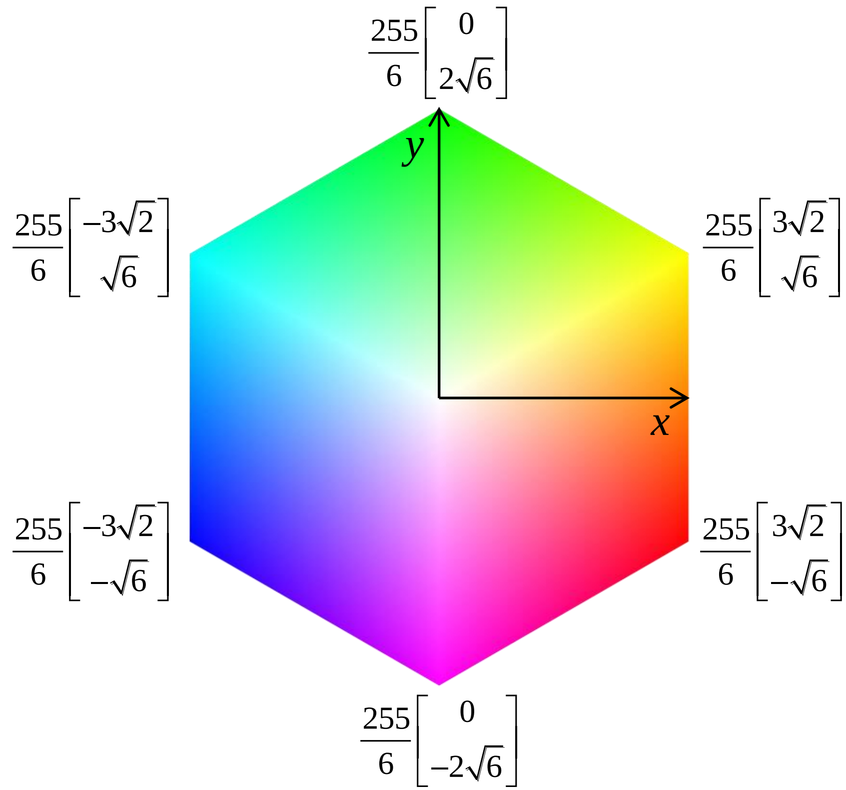
Projected, rotated cube corners:

Projection: set 3rd dim to 0.

$$\Phi\{R\Sigma_{CC}\} = \frac{255}{6}\left\{\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}3\sqrt{2}\\-\sqrt{6}\end{bmatrix},\begin{bmatrix}3\sqrt{2}\\\sqrt{6}\end{bmatrix},\begin{bmatrix}0\\2\sqrt{6}\end{bmatrix},\begin{bmatrix}-3\sqrt{2}\\\sqrt{6}\end{bmatrix},\begin{bmatrix}-3\sqrt{2}\\-\sqrt{6}\end{bmatrix},\begin{bmatrix}0\\-2\sqrt{6}\end{bmatrix},\begin{bmatrix}0\\0\end{bmatrix},\right\}$$

Rotation and Projection of the Color Cube

The origin of this rectangular coordinate system is at the gray point in the center of the hexagon. The y-axis is vertical through the green point; the x-axis is to the right through orange.



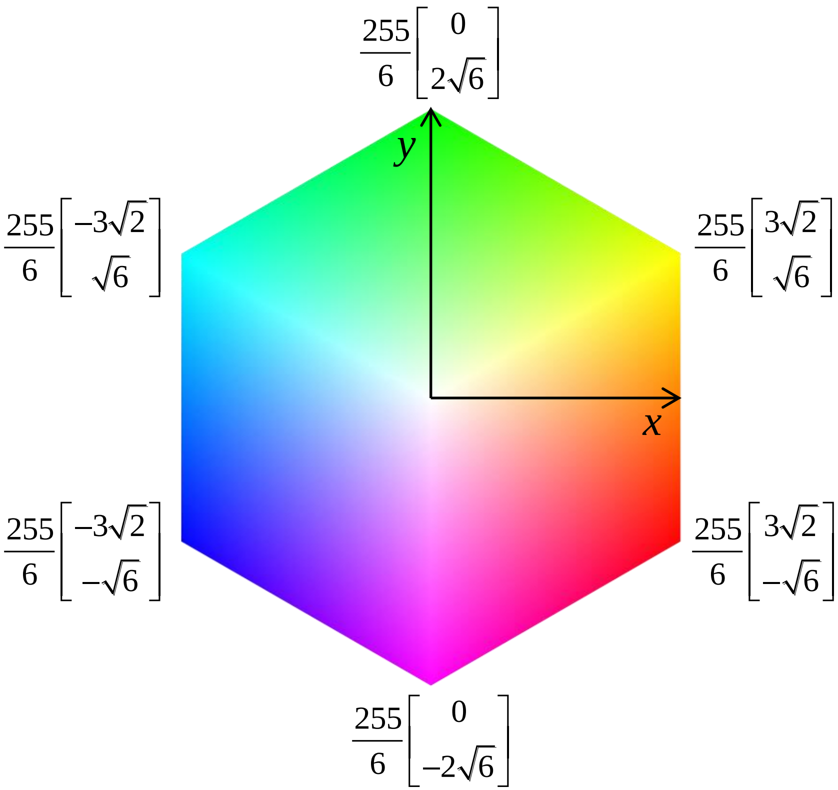
Chromaticity-plane hexagon with red, yellow, green, cyan, blue, and magenta vertices.

Rotation and Projection of the Color Cube

The green axis of the hexagon remains aligned with the green axis of the original color cube, …

The origin of this rectangular coordinate system is at the gray point in the center of the hexagon. The y-axis is vertical through the green point; the x-axis is to the right through orange.

… but the red and blue axes project 60° on either side of green.



Chromaticity-plane hexagon with red, yellow, green, cyan, blue, and magenta vertices.

Geometric Hexagonal Projection

Use color $\mathbf{p}_0 = [128\ \ 50\ \ 206]^T$ as an example. Rotating $\mathbf{p}_0$ with the matrix on slide 106 yields

$$R_{\text{hex}}\mathbf{p}_0 = \frac{1}{6}\begin{bmatrix} 3\sqrt{2} & 0 & -3\sqrt{2} \\ -\sqrt{6} & 2\sqrt{6} & -\sqrt{6} \\ 2\sqrt{3} & 2\sqrt{3} & 2\sqrt{3} \end{bmatrix}\begin{bmatrix} 128 \\ 50 \\ 206 \end{bmatrix} = \begin{bmatrix} -55.1543 \\ -95.5301 \\ 221.7025 \end{bmatrix}.$$

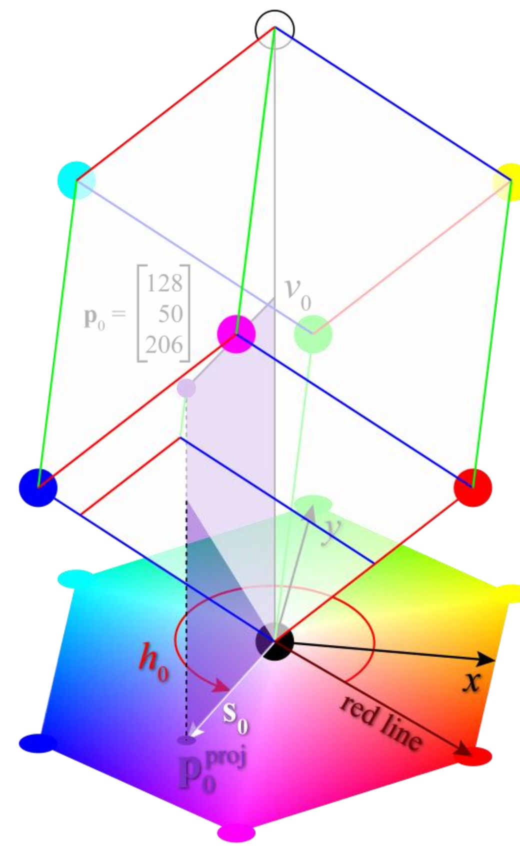
The red axis is -30° from the hexagon's x-axis and $\mathbf{p}_0$'s blue component is larger than its green. Therefore,

$$h_0 = 360 + \operatorname{atan2d}(-95.5301, -55.1543) + 30 \ = 270°.$$

$$v_0 \approx 221.7025\sqrt{3}/3 = 128. \qquad s_{\max} = \tfrac{255}{6}\,2\sqrt{6} \approx 208.2066$$

$$\|\mathbf{s}\| = \left\|\begin{bmatrix} -55.1543 \\ -95.5301 \end{bmatrix}\right\| \approx 110.3807 \qquad s = \frac{\|\mathbf{s}\|}{s_{\max}} \approx 0.5298$$

So we get (h,s,v) = (270°, 0.5298, 128) which is exactly what we got from the geometric algorithms (slides 83 & 86).



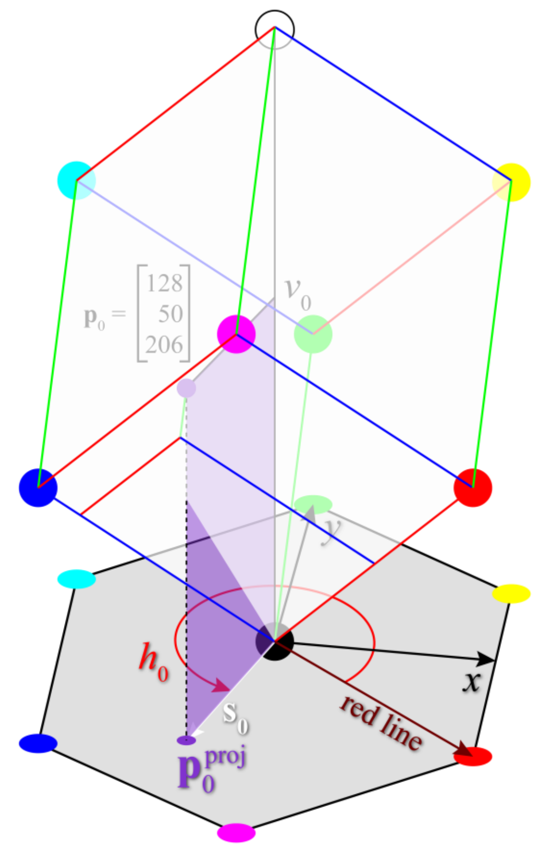
color projection onto the hexagon

Fast Algorithm for RGB to HSV

The fast algorithm on slide 100 uses the hex projection without rotating the color. First the RGB values are normalized by dividing them by the maximum intensity. The test color then scales to

$$\mathbf{p}_0 = \frac{1}{255}\begin{bmatrix}128\\50\\206\end{bmatrix} \approx \begin{bmatrix}0.5020\\0.1961\\0.8078\end{bmatrix}.$$

The value of the color is taken to be simply the maximum of the normalized rgb values. Therefore $v_0 = 0.8078$, whereas the geometric algorithm, if normalized would have yielded $v_0 = 0.5$.



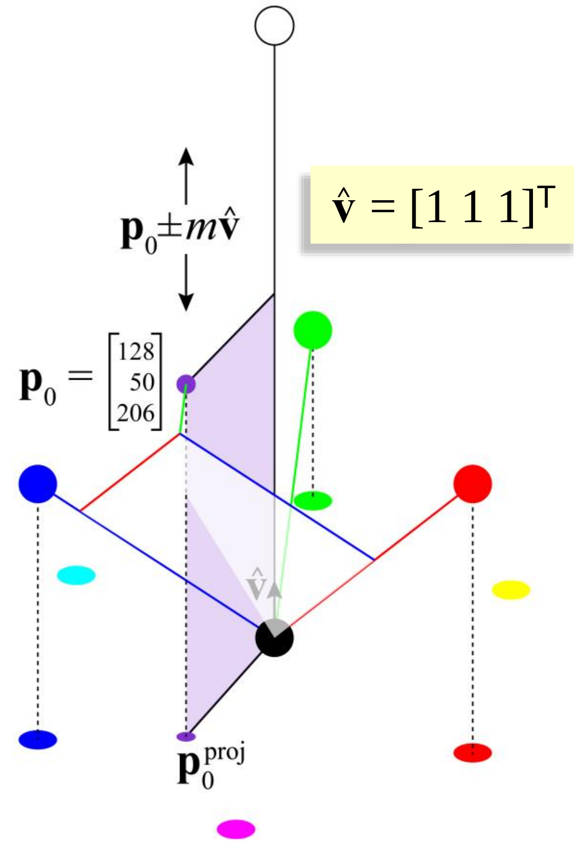
color projection onto the hexagon

Fast Algorithm for RGB to HSV

To project the color onto the hexagon the fast algorithm subtracts the minimum of the rgb intensities from all three. The reasoning is:

Let $\mathbf{p}_0' = \mathbf{p}_0 \pm m\hat{\mathbf{v}}$ for $\hat{\mathbf{v}}$, the unit vector in the direction of the gray line and m a real number. The value of m shifts $\mathbf{p}_0$ up or down in the direction of the gray line. The result projects to the same point, $\mathbf{p}_0^{\text{proj}}$, in the hexagon independent of the value of m.

If m is min(r,g,b), then the color is shifted to one of the faces of the original color cube.



color projection onto the hexagon

Fast Algorithm for RGB to HSV

For our test color, $m = \min\{r, g, b\}$

$$= \min\{0.5020,\ 0.1961,\ 0.8078\}$$

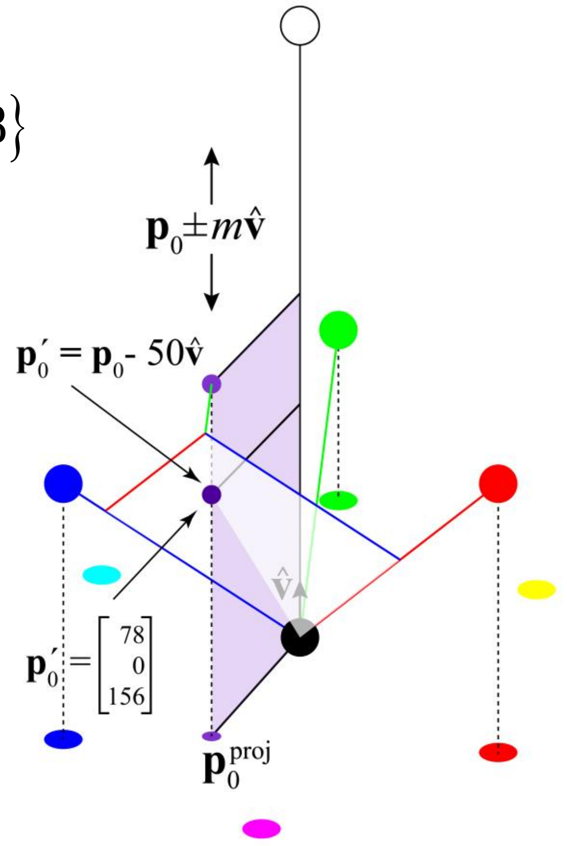
$$= 0.1961.$$

Subtract m from $\mathbf{p}_0$ to get

$$\mathbf{p}_0' = \mathbf{p}_0 - m\hat{\mathbf{v}}$$

$$= \begin{bmatrix} 0.5020 \\ 0.1961 \\ 0.8078 \end{bmatrix} - 0.1961 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.3059 \\ 0 \\ 0.6118 \end{bmatrix},$$

which is in the *BR*-plane of the color cube, since $G = 0$. (See slide 100.) Note that $\mathbf{p}_0'$ projects to the same point on the hexagon as does $\mathbf{p}_0$.



color projection onto the hexagon

Fast Algorithm for RGB to HSV

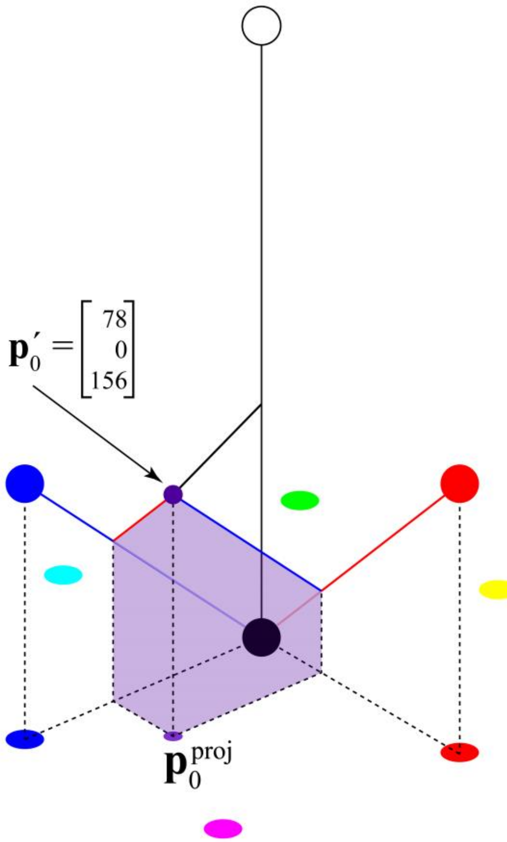
Define the *chroma* as $C = M - m$. This definition is justified by noting that

1. Chroma $C = 0$ for any color on the gray line
2. any color with two zero components is a primary,
3. for any color with one zero component the formula makes C equal the maximum of the two nonzero values.

Hue is defined with respect to chroma as

$$H' = \begin{cases} 0, & \text{if} \quad C = 0, \\ \frac{G-B}{C} \bmod 6, & \text{if} \quad M = R, \\ \frac{B-R}{C} + 2, & \text{if} \quad M = G, \\ \frac{R-G}{C} + 4, & \text{if} \quad M = B. \end{cases}$$

$$H = 60° \times H'$$



color projection onto the hexagon

Fast Algorithm for RGB to HSV

So, for $\mathbf{p}_0 = \frac{1}{255}\begin{bmatrix}128\\50\\206\end{bmatrix}$ & $v_0 = \frac{206}{255} \approx 0.8078,$

We get

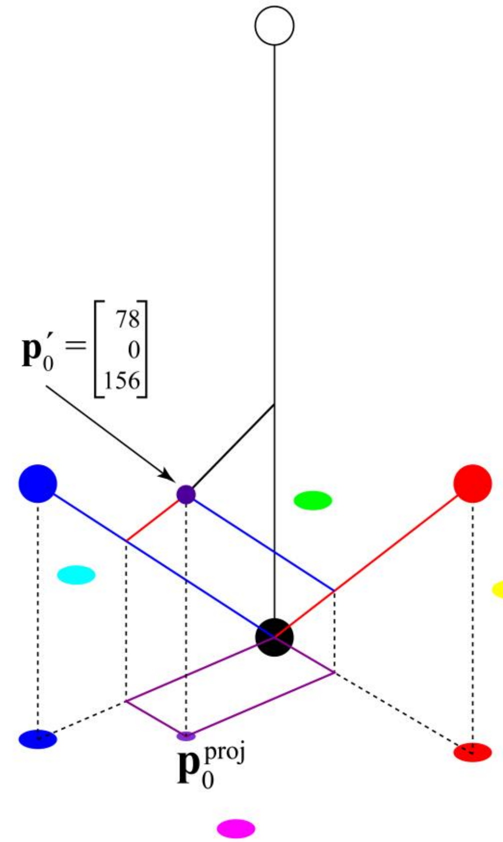
$$M = \frac{B}{255} = \frac{206}{255} \approx 0.8078$$

$$m = \frac{G}{255} = \frac{50}{255} \approx 0.1961,$$

$$C = \frac{M-m}{255} = \frac{206-50}{255} = \frac{156}{255} \approx 0.6118,$$

$$H' = \frac{R-G}{C} + 4 = \frac{128-50}{156} + 4 = \frac{1}{2} + 4 = \frac{9}{2},$$

$$H = 60^\circ \times H' = 60^\circ \times \frac{9}{2} = 270^\circ.$$



which is exactly the hue from the geometric algorithms.

color projection onto the hexagon

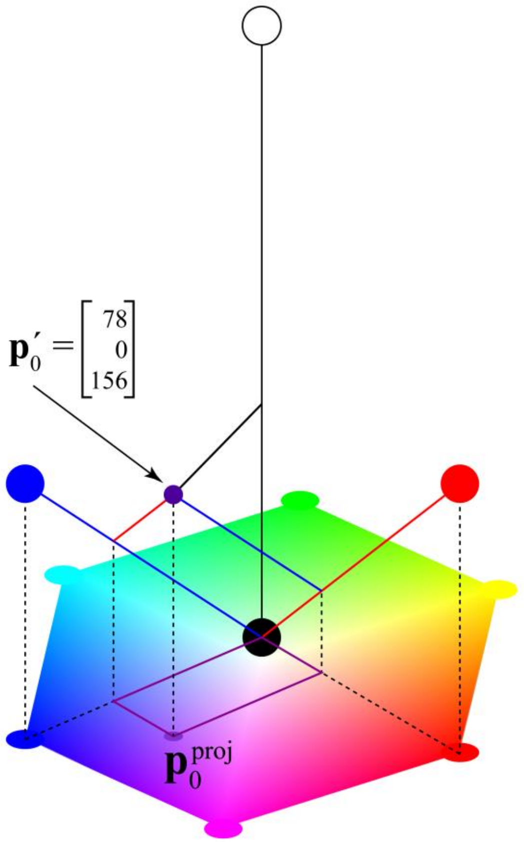
Fast Algorithm for RGB to HSV

Value is defined as either

$$V = \tfrac{1}{3}\left(R+G+B\right) \quad \text{or} \quad V = \max(R,G,B).$$

Saturation is defined with respect to chroma and value as

$$S = \begin{cases} 0, & \text{if} \quad V = 0 \\ 1-\dfrac{m}{V}, & \text{if} \quad V \neq 0 \end{cases}, \quad \text{or} \quad S = \begin{cases} 0, & \text{if} \quad V = 0 \\ \dfrac{C}{V}, & \text{if} \quad V \neq 0 \end{cases}.$$



color projection onto the hexagon

Fast Algorithm for RGB to HSV

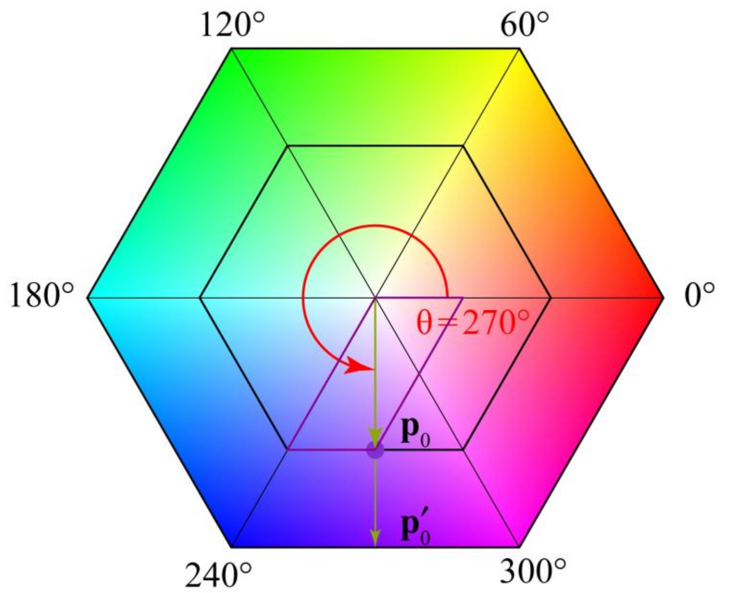
Therefore with the color $\mathbf{p}_0$ we have

$$\begin{aligned} V &= \tfrac{1}{3}\left(R+G+B\right) \\ &= \tfrac{1}{765}\left(128+50+206\right) = \tfrac{384}{765} \approx 0.5020, \quad \text{or} \\ V &= \max(R,G,B) \\ &= \tfrac{206}{255} \approx 0.8078. \end{aligned}$$

Saturation is defined with respect to chroma and value as

$$S = 1 - \frac{m}{V} = 1 - \frac{50/255}{384/765} \approx 0.6094, \quad \text{or}$$

$$S = \frac{C}{V} = \frac{156/255}{206/255} = \frac{156}{206} \approx 0.7573.$$



Hue and Saturation via Projection onto the Hexagon

For color $\mathbf{p}_0 = [128 \;\; 50 \;\; 206]^\mathsf{T}$ we have three different HSV triples,

$$\begin{bmatrix} 128 \\ 50 \\ 206 \end{bmatrix}$$

by vector math,

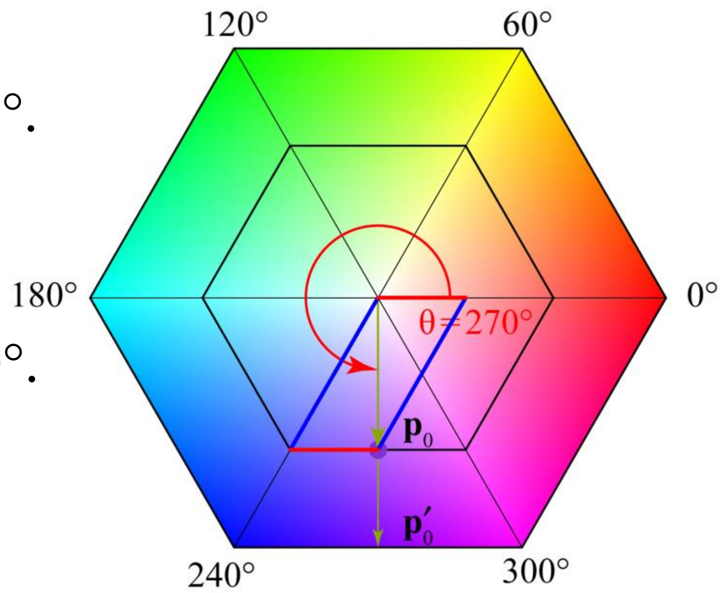
$$v_0 = 0.5000, \quad s_0 = 0.5298, \quad \text{and} \quad h_0 = 270°.$$

by the first hexagon projection,

$$v_0 = 0.5020, \quad s_0 = 0.6094, \quad \text{and} \quad h_0 = 270°.$$

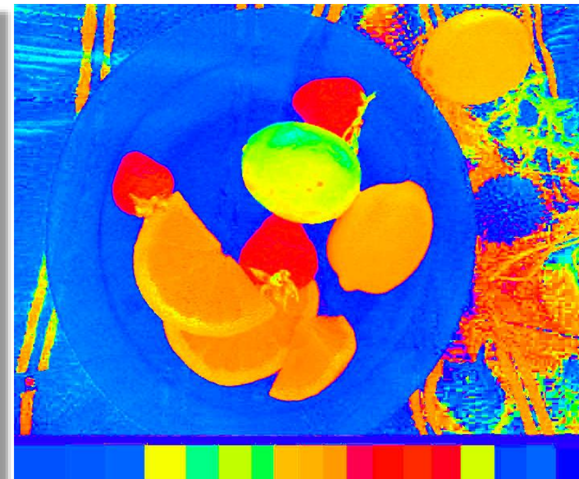
and by the second hexagon projection,

$$v_0 = 0.8078, \quad s_0 = 0.7573, \quad \text{and} \quad h_0 = 270°.$$



Example RGB Image and HSV Bands

Original image Karen Gillis Taylor, “A Fruit Color Study.”



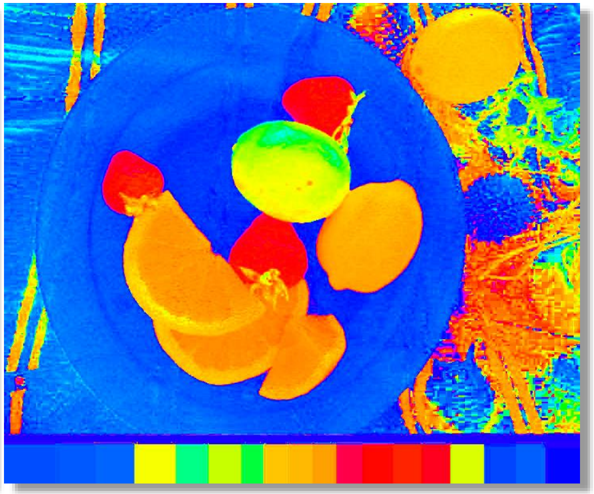
Hue image displayed with a primary color color-map

Value Image

Saturation image scaled to (0,…,255)

Example RGB Image and HSV Bands

Original image Karen Gillis Taylor, “A Fruit Color Study.”



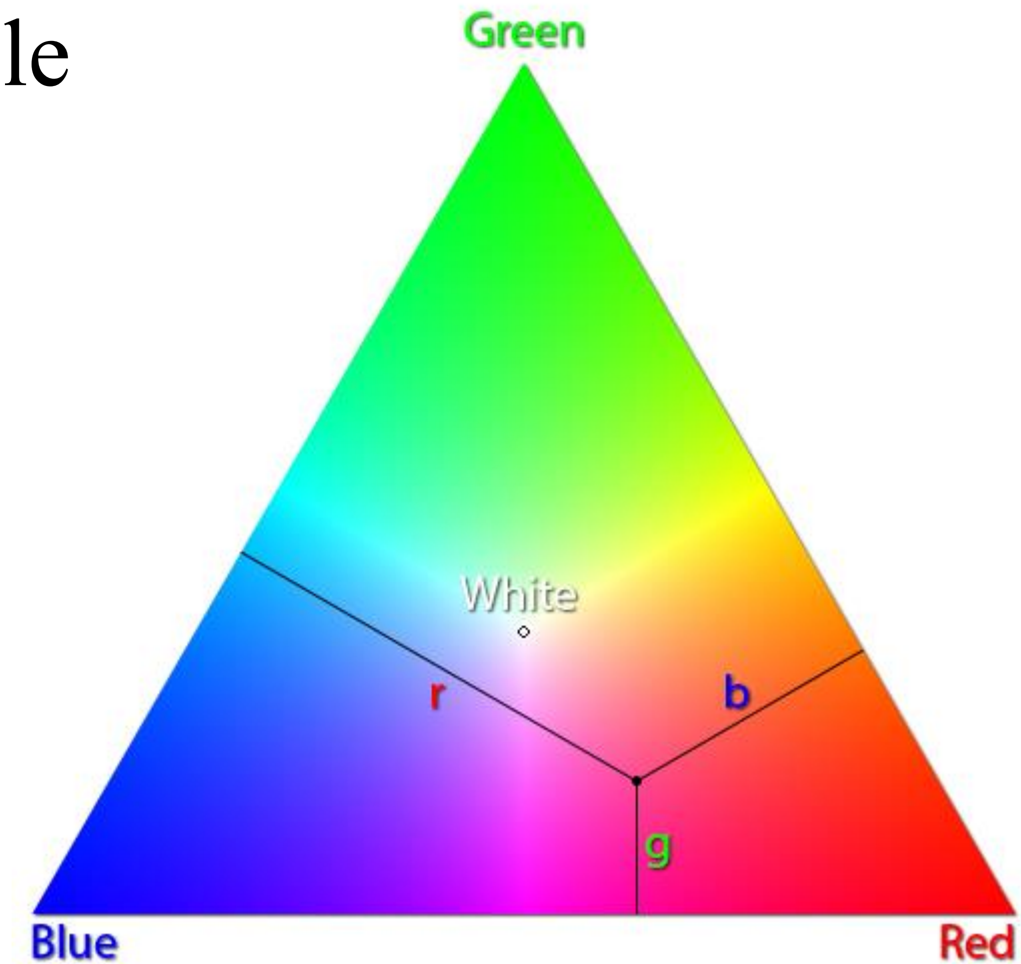
Hue image displayed with a primary color color-map

Value Image

Saturation image scaled to (0,…,255)

Maxwell's Triangle

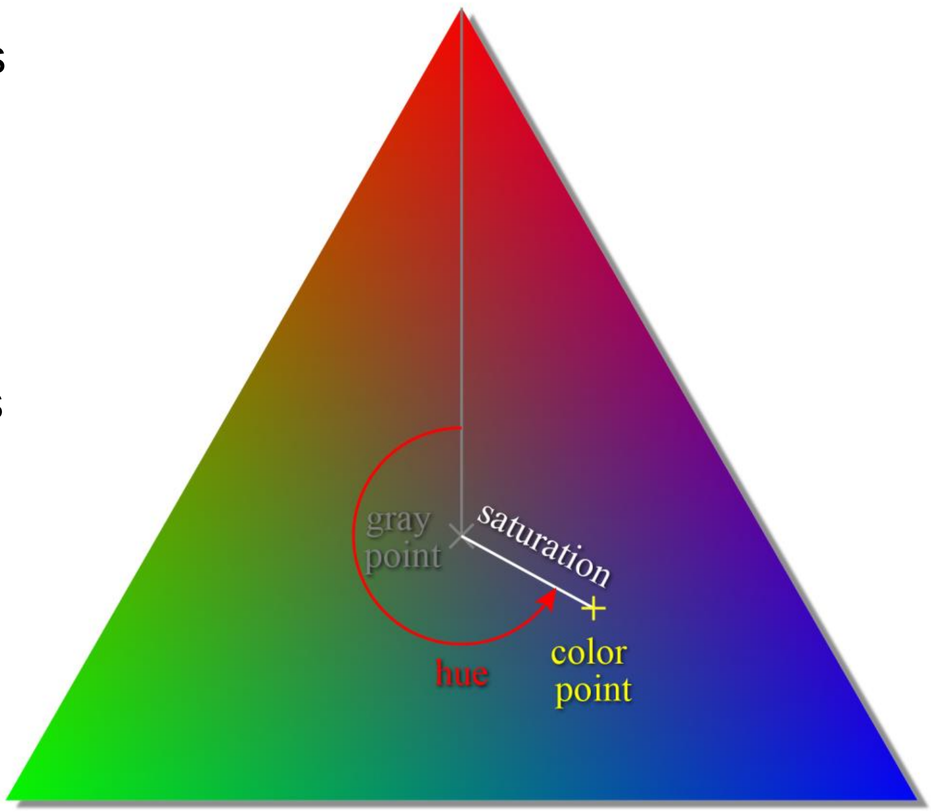
Probably the first attempts to produce color curves describing the trichromatic theory of color were those by Maxwell (1857, 1860).… [The] first chromaticity diagram was a circle devised by Newton. Later, Maxwell used an equilateral triangle.… In his trichromatic theory, each of the three primary colors—red, green, and blue—is located at a corner of the triangle. The white color is in the middle. Other colors are formed by a combination of the r, g, b components depending on the distances from each of the three sides of the triangle. This triangular representation has been used often with several modifications.



D. Malacara-Hernandez, Color Vision and Colorimetry: Theory and Applications, SPIE Press, (2002).

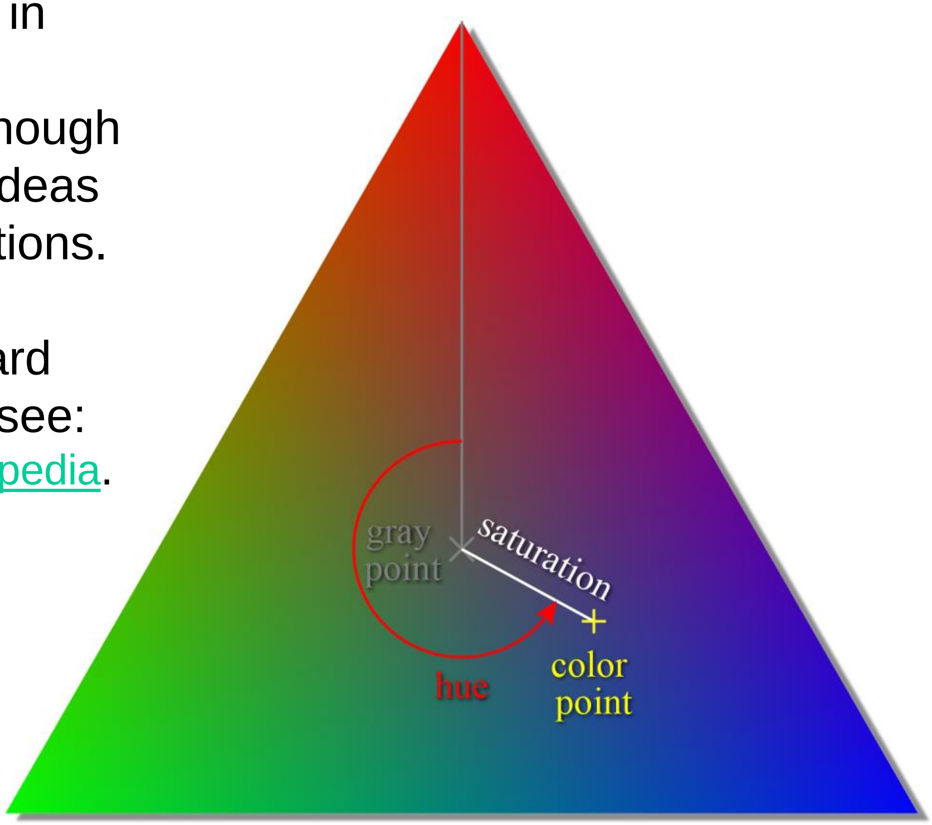
Brightness + Chrominance Representation

There are many different ways to encode color in terms of 1D brightness and 2D chrominance. Chrominance is often represented in terms of hue and saturation. A given brightness measure (*e.g*. value or NTSC luminance) defines a planar surface in the color cube on which the brightness is constant. One point on that surface is gray. The saturation of any color with the given brightness is defined as the distance on the plane from the color to the gray point. The hue is defined as the angular deviation from red measured in the same plane.



Brightness + Chrominance Representation

The HSV encoding scheme presented in the first part of this lecture is a direct implementation of the vector math. Although it is nonstandard, it demonstrates the ideas that underlie most of these representations.



For a good explanation of more standard HSV and LHS representations please see: HSL and HSV - Wikipedia, the free encyclopedia.

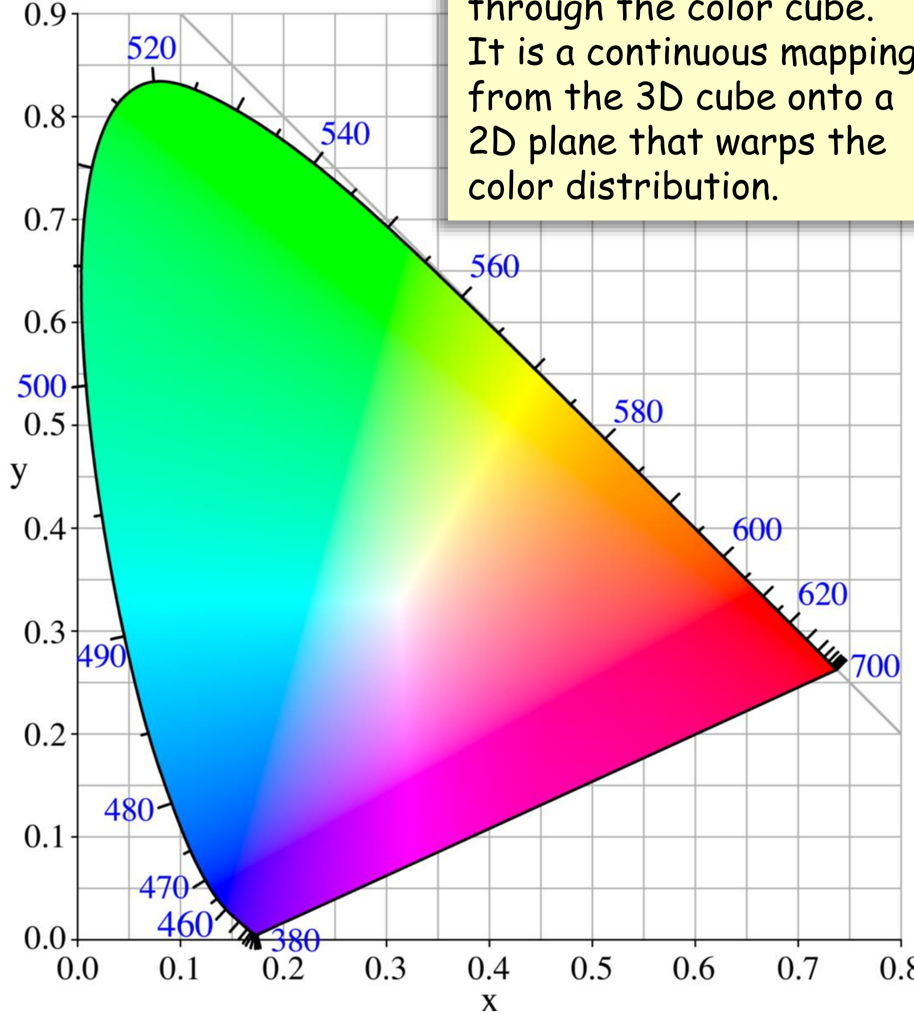
An explanation of the hexagonal representation of HSV is given later in this lecture.

C.I.E. 1931 Color Space

Commission Internationale de l’Éclairage Uniform Color Scale

“The CIE system characterizes colors by a luminance parameter Y and two color coordinates x and y which specify the point on the chromaticity diagram. This system['s] ... parameters are based on the spectral power distribution ... of the light emitted from a colored object and are factored by sensitivity curves which have been measured for the human eye.”[1]

This is not a simple slice through the color cube. It is a continuous mapping from the 3D cube onto a 2D plane that warps the color distribution.



[1] http://hyperphysics.phy-astr.gsu.edu/hbase/vision/cie.html

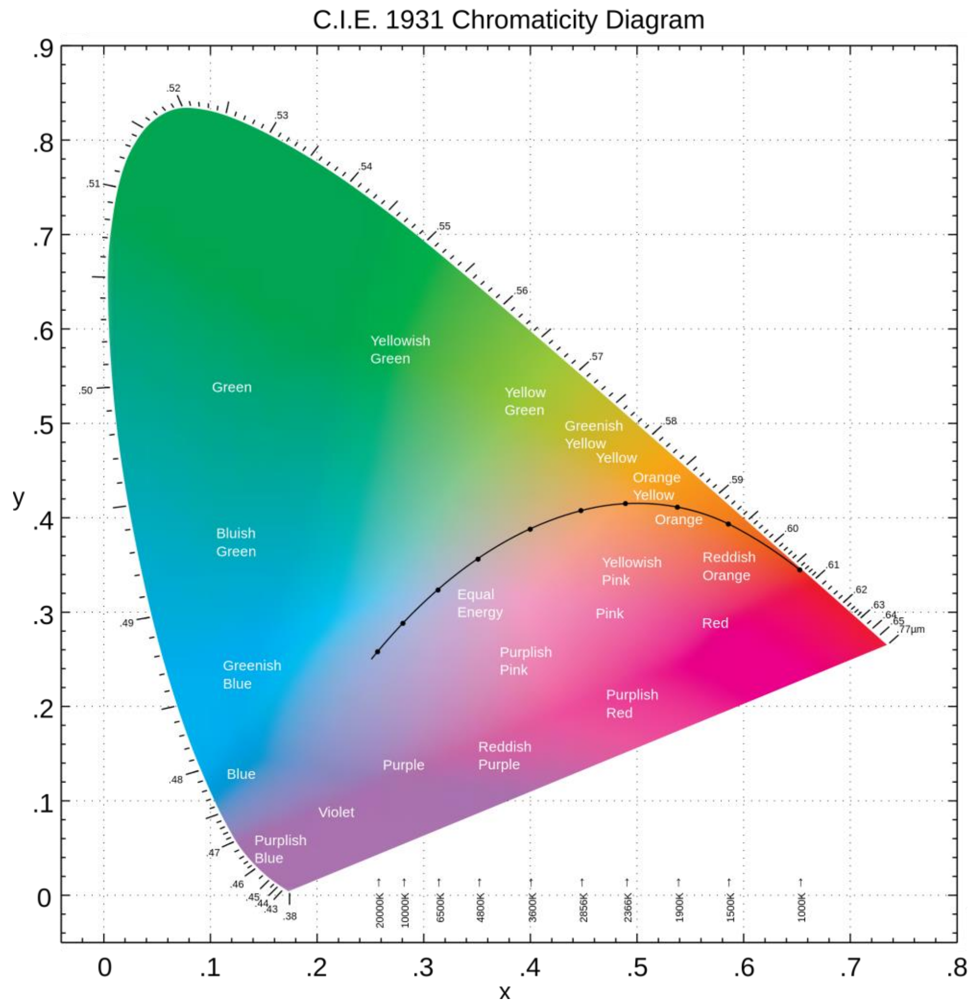
C.I.E. 1931

This 1931 chromaticity diagram shows a single xy plane in the xyY color space.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z}$$

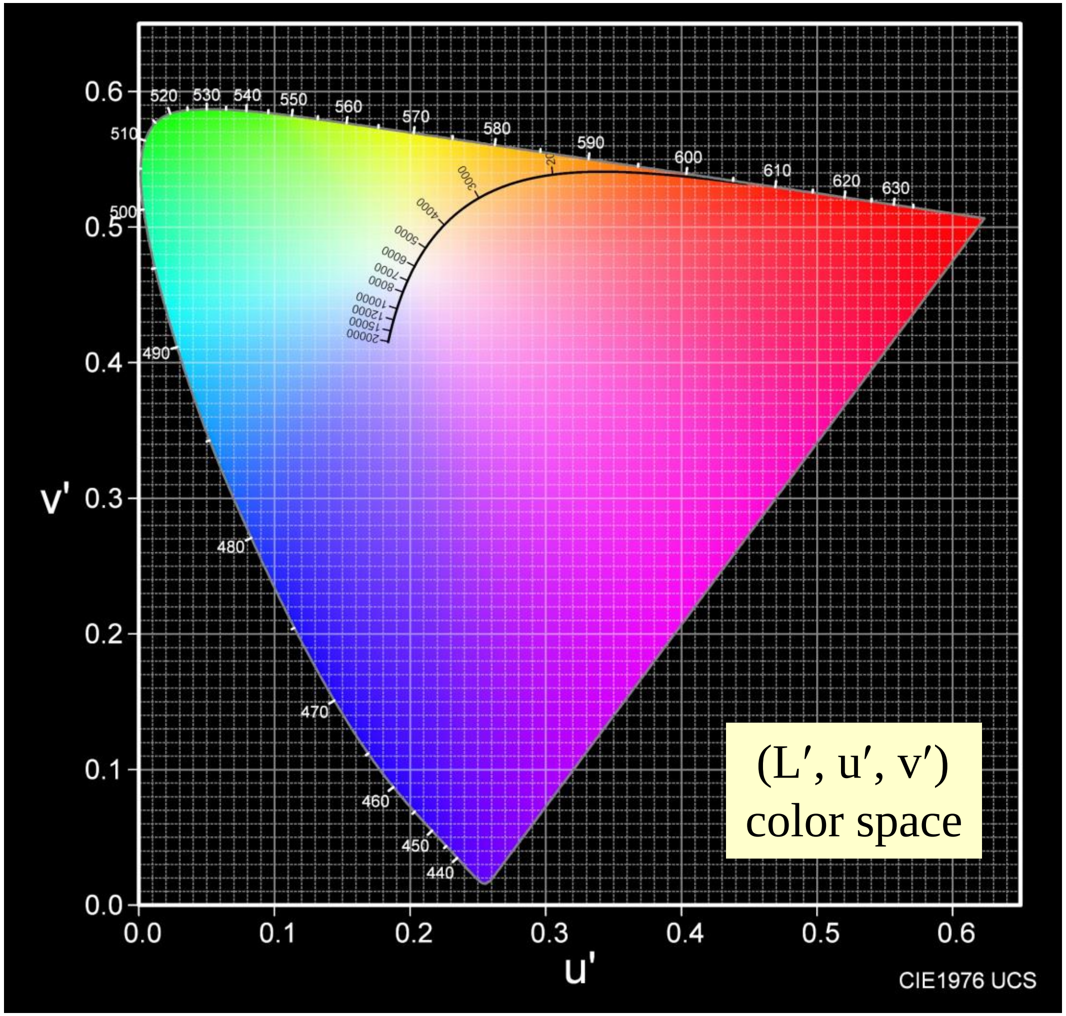
"The CIE 1931 color space chromaticity diagram rendered in terms of the colors of lower saturation and value than those displayed in the diagram above that can be produced by pigments, such as those used in printing. The color names are from the Munsell color system."[2,3]



[2] https://en.wikipedia.org/wiki/CIE_1931_color_space
[3] https://en.wikipedia.org/wiki/Munsell_color_system

C.I.E. 1976 U.C.S.

This diagram, produced by the C.I.E. in 1976 specifies the range of colors perceptible at a single luminance, L′, by an average person. On the periphery the colors are monochromatic colors labeled with the corresponding wave-lengths of light. The white point is intersected by a curve that displays the corresponding black body temperature. (u′, v′) are the CIE chromaticity coordinates. The distance between points is related to the perceptual difference in color.



http://hyperphysics.phy-astr.gsu.edu/hbase/vision/cie1976.html

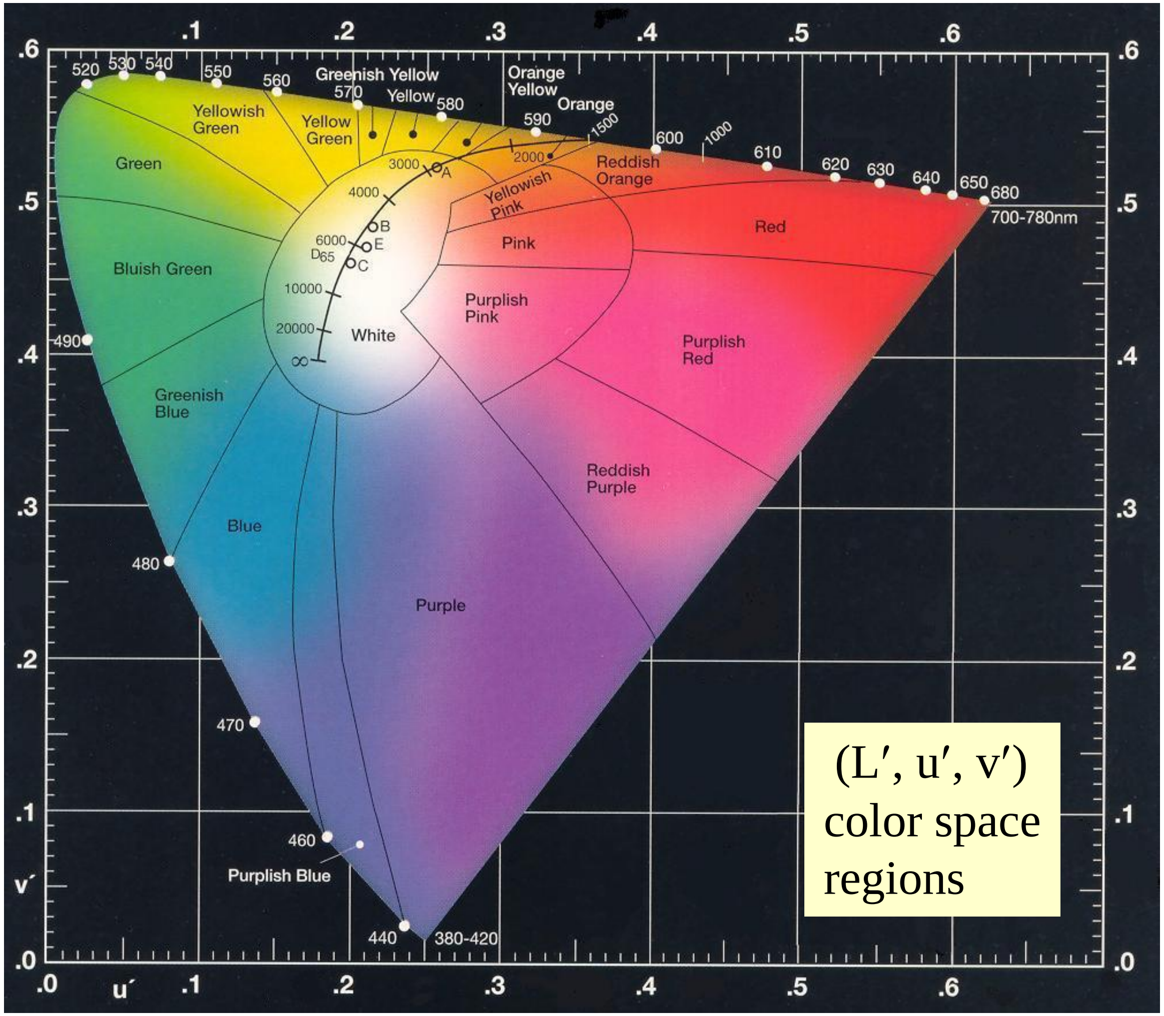
C.I.E. 1976 U.C.S.

Color regions on a slightly desaturated CIE 1976 L′u′v′ color space similar to those on the CIE 1931 xyY digram on slide [125](#).

Point $\mathbf{c}_0$ on the line between colors $\mathbf{c}_1$ and $\mathbf{c}_1$ is the color that would be obtained by mixing the two endpoint colors in proportion to the distances from the endpoints. *E.g.*

$$\mathbf{c}_0 = \alpha\mathbf{c}_1 + (1-\alpha)\mathbf{c}_2$$

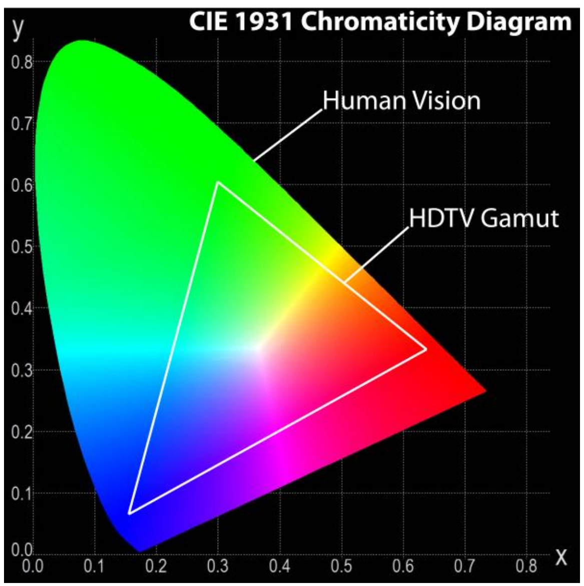
where $0 \le \alpha \le 1$.

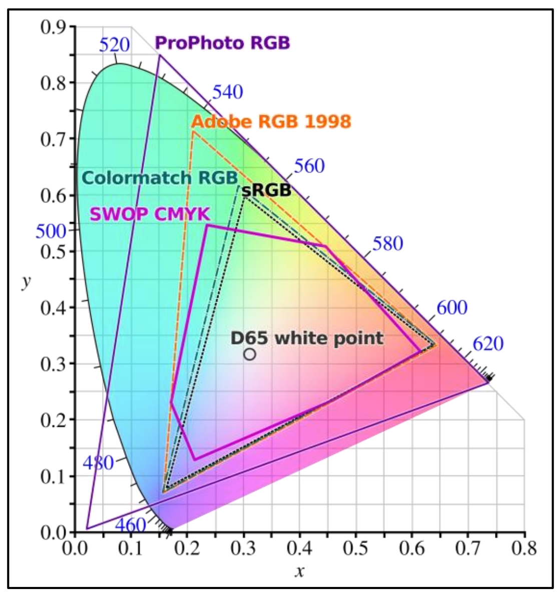


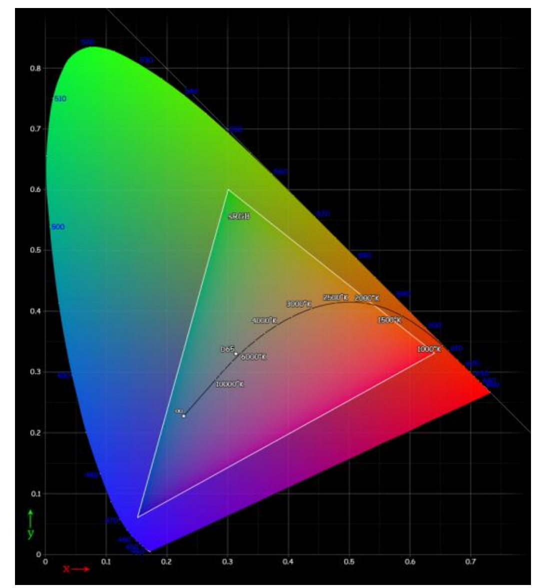
http://hyperphysics.phy-astr.gsu.edu/hbase/vision/cie1976.html

Color Gamuts

A color *gamut* is the range of colors that can be represented by a given display. The pure R, G, and B values are plotted on a CIE chromaticity diagram. The colors inside the triangle are the ones that can be displayed by the device.



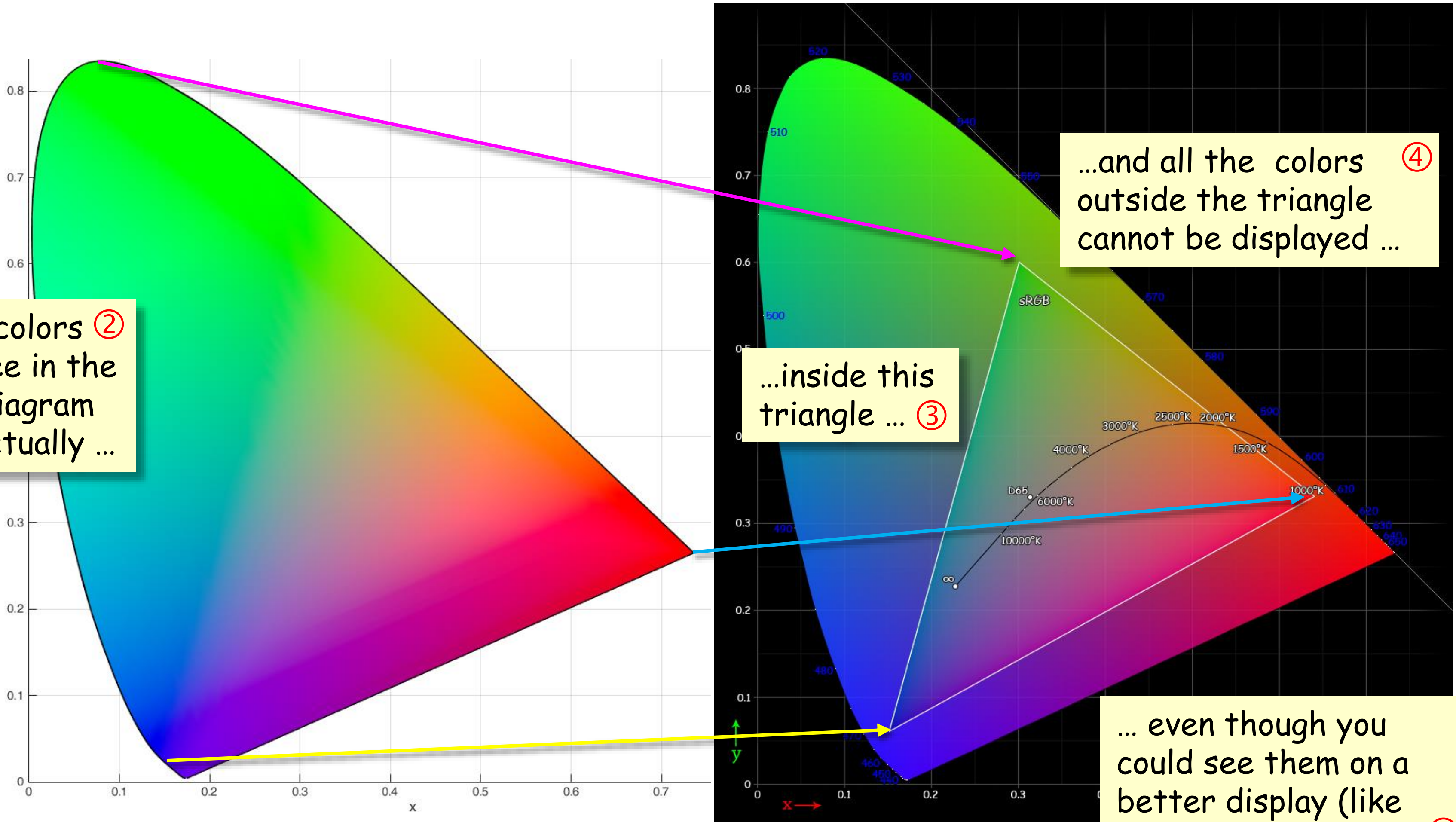




Since the display you are looking at is probably sRGB … ①

Color Gamuts

…the colors ② you see in the CIE diagram are actually …



…inside this triangle … ③

…and all the colors ④ outside the triangle cannot be displayed …

… even though you could see them on a better display (like reality!) ⑤