Converting from RGB to HSV

Color vision can be processed using RGB color space or HSV color space. RGB color space describes colors in terms of the amount of red, green, and blue present. HSV color space describes colors in terms of the **H**ue, **S**aturation, and **V**alue. In situations where color description plays an integral role, the HSV color model is often preferred over the RGB model. The HSV model describes colors similarly to how the human eye tends to perceive color. RGB defines color in terms of a combination of primary colors, where as, HSV describes color using more familiar comparisons such as color, vibrancy and brightness. The basketball robot uses HSV color space to process color vision.

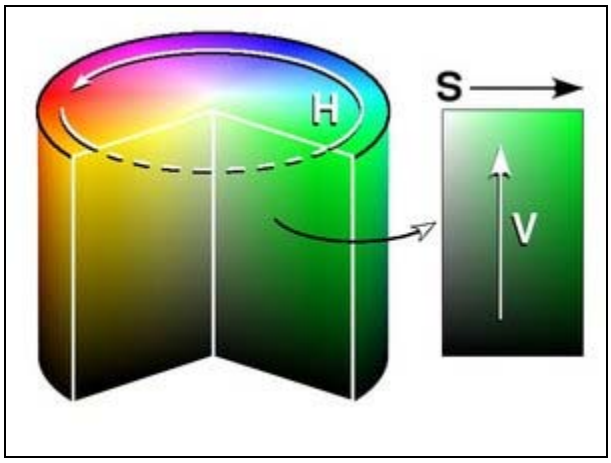

Figure 1: HSV color wheel

Figure 1 illustrates how hue, saturation, and value are defined.

- Hue represents the color type. It can be described in terms of an angle on the above circle. Although a circle contains 360 degrees of rotation, the hue value is normalized to a range from 0 to 255, with 0 being red.
- Saturation represents the vibrancy of the color. Its value ranges from 0 to 255. The lower the saturation value, the more gray is present in the color, causing it to appear faded.
- Value represents the brightness of the color. It ranges from 0 to 255, with 0 being completely dark and 255 being fully bright.
- White has an HSV value of 0-255, 0-255, 255. Black has an HSV value of 0-255, 0-255, 0. The dominant description for black and white is the term, value. The hue and saturation level do not make a difference when value is at max or min intensity level.

The color camera, on the robot, uses the RGB model to determine color. Once the camera has read these values, they are converted to HSV values. The HSV values are then used in the code to determine the location of a specific object/color for which the robot is searching. The pixels are individually checked to determine if they match a predetermined color threshold.

The generic C-code for converting RGB color to HSV color is given below.

```
// r,g,b values are from 0 to 1
// h = [0,360], s = [0,1], v = [0,1]
//              if s == 0, then h = -1 (undefined)
void RGBtoHSV( float r, float g, float b, float *h, float *s, float *v )
{
        float min, max, delta;
        min = MIN( r, g, b );
        max = MAX( r, g, b );
        *v = max;                               // v
        delta = max - min;
        if( max != 0 )
                *s = delta / max;               // s
        else {
                // r = g = b = 0                // s = 0, v is undefined
                *s = 0;
                *h = -1;
                return;
        }
        if( r == max )
                *h = ( g - b ) / delta;                 // between yellow &
magenta
        else if( g == max )
                *h = 2 + ( b - r ) / delta;     // between cyan & yellow
        else
                *h = 4 + ( r - g ) / delta;     // between magenta & cyan
        *h *= 60;                               // degrees
        if( *h < 0 )
                *h += 360;
}
```