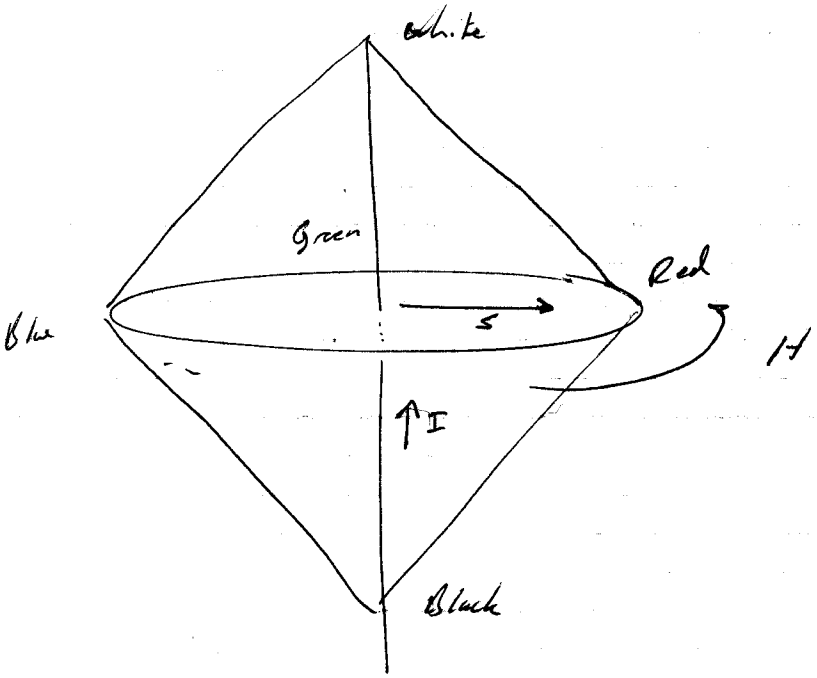
Color Schemes



I gives intensity. When I is close to Black or White, you cannot trust H and S.

H indicates the colors and should provide fairly effective color segmentation

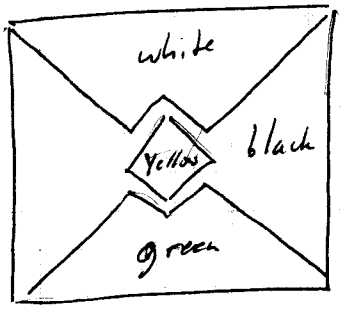
S indicates the purity of the color

Feature Detection

As an alternative to color segmentation, you may want to add feature detection capability to make the system lighting invariant.

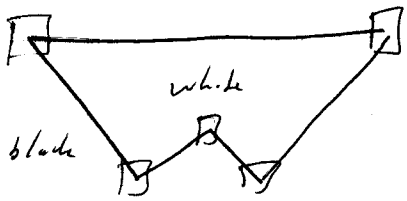
Basic idea:

Given the following dot pattern

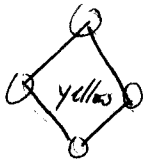


- If the intensity is at one of the extremes then yellow and green may appear as different colors ~~any~~ and get missed by the color segmentation algorithm.
- when this happens (detected by thresholding on I)
- I may be better to find corners between high contrasting colors

Eg



or



Question: How do we detect corners?

Step 1: For every pixel in area of interest, compute the gradients I_x and I_y

- This is most easily done with the mask

-1	0	1
-2	0	2
-1	0	1

I_x

1	2	1
0	0	0
-1	-2	-1

I_y

or

-1	0	0	0	1
0	0	0	0	0
-2	0	0	0	2
0	0	0	0	0
-1	0	0	0	1

I_x

1	0	2	0	-1
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
-1	0	-2	0	-1

I_y

Step 2

For each pixel $\underset{\sim}{I}$ in area of interest, compute

a) the matrix

$$G = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}$$

where the sum is over a subwindow centered at I

b) the quantity

$$C = \det G + k \, \text{trace}^2 G$$
$$= \left(\sum I_x^2\right)\left(\sum I_y\right)^2 - \left(\sum I_x I_y\right)^2 + k\left(\sum I_x^2 + \sum I_y^2\right)$$

Step 3 Segment corners based on C.

- corners will have large value of C.