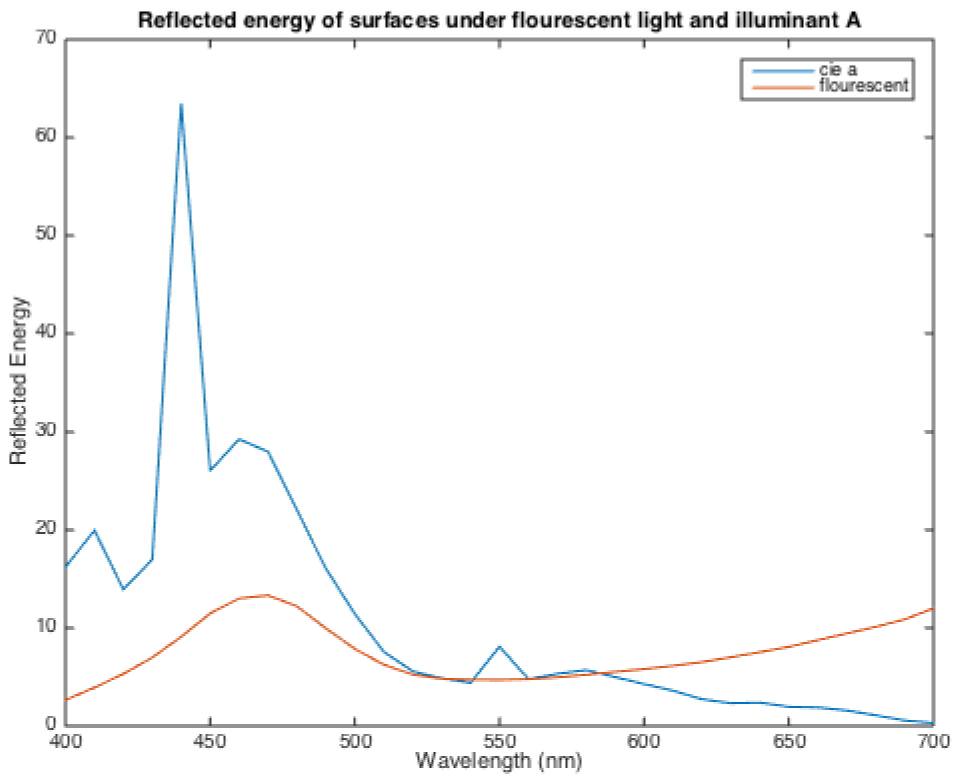
PERCEPTION COLOR ASSIGNMENT

1) Making a graph of the 18th Macbeth surface

```
spectrum = linspace(400,700,31);
load surfaces
load illuminants
spect18fl = macbeth(18,:)' .* flourescent';
spect18a = macbeth(18,:)' .* cie_a';
plot(spectrum,[spect18fl, spect18a]);
xlabel('Wavelength (nm)');
ylabel('Reflected Energy');
title('Reflected energy of surfaces under flourescent light and illuminant A')
legend('cie a','flourescent');
```




Cone responses under flourescent light

```
load cones
coneSignals18fl = cones * spect18fl
```

```
coneSignals18fl =

   66.1124
   68.4875
  182.9654
```

Cone responses under illuminant A

```
coneSignals18a = cones * spect18a


coneSignals18a =

   65.2373
   53.5643
   61.9800
```

Flourescent looks more blueish because S cells are much more active with little change in L and M

2) Computing the monitor intensities for color matching under illuminant A

```
load phosphors
monitor_to_cones = cones * phosphors';
cones_to_monitor = inv(monitor_to_cones);
monitorSignals = cones_to_monitor * coneSignals18a


monitorSignals =

  309.0330
  192.4875
  415.4957
```

There are intensities greater than 255 so cannot be displayed on this monitor

3) Cone responses to baseline stimulus

```
baseline = phosphors' * [1 1 1]';
baseline_cones =  cones * baseline


baseline_cones =

    0.2638
    0.2257
    0.1591
```

Monitor signal required to increase S cone response by .5

```
coneSignalsDeltaS = baseline_cones + [0 0 0.5]'; %increment S
monitorSignalsDeltaS = cones_to_monitor * coneSignalsDeltaS %


monitorSignalsDeltaS =

    1.4819
    0.3054
    4.6888
```

If we increment by .8, we get a negative monitor intensity which is impossible

```
coneSignalsDeltaS = baseline_cones + [0 0 0.8]'; %increment S
monitorSignalsDeltaS = cones_to_monitor * coneSignalsDeltaS
```

```
monitorSignalsDeltaS =

    1.7710
   -0.1113
    6.9021
```

.7 seems to be highest possible increment (rounded to the nearest tenth)

```
coneSignalsDeltaS = baseline_cones + [0 0 0.7]'; %increment S
monitorSignalsDeltaS = cones_to_monitor * coneSignalsDeltaS
```

```
monitorSignalsDeltaS =

    1.6746
    0.0276
    6.1643
```

4) Computing color matching function

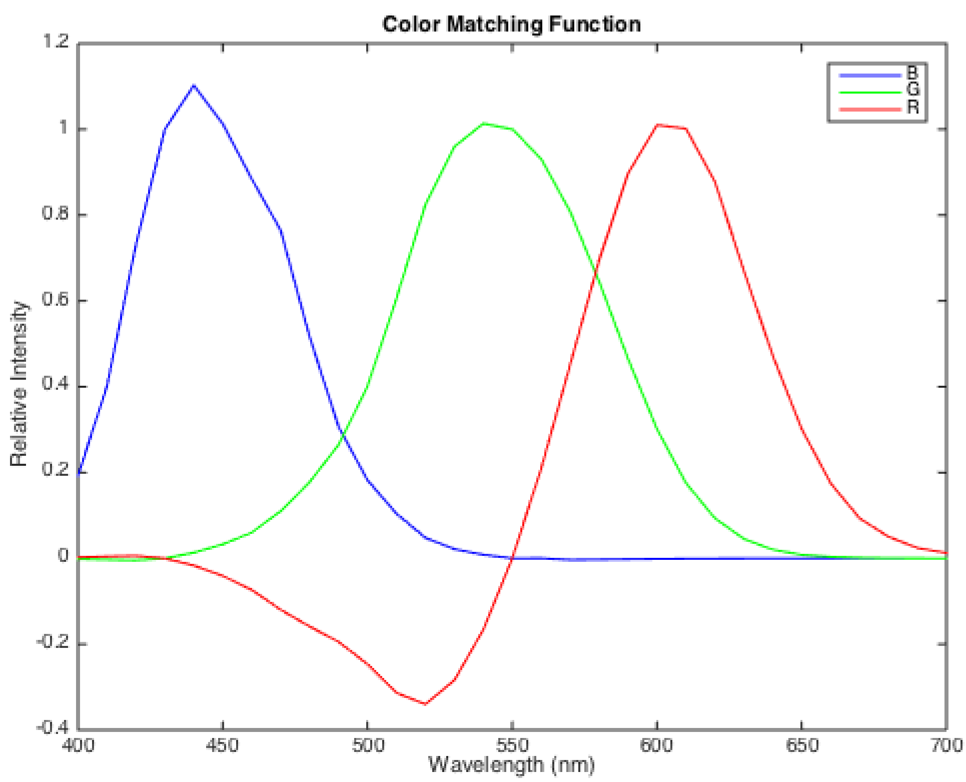
Given a test light %t%, we can figure out the intensities $e_{cie}$ of the cie lights needed to match as $Ct = CM_{cie}e_{cie}$

Rewritten, this becomes $(CM_{cie})^{-1}Ct = e_{cie}$ so the color matching function is $(CM_{cie})^{-1}C$

```
cie = zeros(3, 31);
cie(sub2ind(size(cie), [1 2], [4 16])) = 1;
cie(3, 31) = 80;
color_matching_function = inv(cones * cie') * cones
plot(spectrum,color_matching_function(1,:),'b');
hold on
plot(spectrum,color_matching_function(2,:),'g');
plot(spectrum,color_matching_function(3,:),'r');
xlabel('Wavelength (nm)');
ylabel('Relative Intensity');
title('Color Matching Function')
legend('B','G','R');
```

```
color_matching_function =

  Columns 1 through 7

    0.1923    0.4017    0.7308    1.0000    1.1025    1.0127    0.8844
   -0.0016   -0.0031   -0.0044   -0.0000    0.0133    0.0327    0.0602
    0.0024    0.0045    0.0059   -0.0000   -0.0167   -0.0409   -0.0734

  Columns 8 through 14
```

```
   0.7645    0.5163    0.3066    0.1820    0.1042    0.0478    0.0215
   0.1100    0.1779    0.2647    0.4010    0.6061    0.8244    0.9595
  -0.1198   -0.1588   -0.1945   -0.2467   -0.3136   -0.3400   -0.2840

 Columns 15 through 21

   0.0085   -0.0000    0.0003   -0.0035   -0.0028   -0.0020   -0.0013
   1.0135    1.0000    0.9311    0.8081    0.6455    0.4651    0.3013
  -0.1664   -0.0000    0.2093    0.4512    0.6947    0.8970    1.0100

 Columns 22 through 28

  -0.0007   -0.0004   -0.0002   -0.0001   -0.0000   -0.0000   -0.0000
   0.1754    0.0934    0.0453    0.0202    0.0082    0.0036    0.0014
   1.0017    0.8779    0.6706    0.4723    0.3012    0.1750    0.0932

 Columns 29 through 31

  -0.0000   -0.0000   -0.0000
   0.0005    0.0003    0.0000
   0.0503    0.0242    0.0125
```




5) Given a vector of cie intensities, we can compute the requisite phosphor intensities withouth reference to a test SPD. This is because we can write the SPD of a set of cie intensities as $t = M_{cie} e_{cie}$

So the above equation for the phosphors becomes $(CM_{phos})^{-1}CM_{cie}e_{cie} = e_{phos}$ which gives us $(CM_{phos})^{-1}CM_{cie}$ as the function to go from cie intensities to phosphor intensities

```
cie_to_cones = cones * cie';
cones_to_phosphors = inv(cones * phosphors');
cie_to_phosphors = cones_to_phosphors * cie_to_cones
```

```
cie_to_phosphors =

    0.7444    0.0655    8.8624
   -1.0309    6.1927   -1.9196
    6.6693   -0.6997   -0.0999
```

*Published with MATLAB® R2014b*