

CMSC 828D: Fundamentals of Computer Vision Homework 2

Instructors : Larry Davis, Ramani Duraiswami,
Daniel DeMenthon, and Yiannis Aloimonos

1. A surface of 1 square meter receives around 1000 Watts of solar power on a sunny day. Use the average wavelength of the solar spectrum and the energy of a photon at that wavelength to calculate a rough estimate of the number of photons the surface receives per second. (See lecture notes for any data needed)

Solution: One photon has energy

$$Q = hf = hc/\lambda$$

where $h = 6.623 \times 10^{-34}$ J s is Planck's constant, $c = 2.998 \times 10^8$ ms⁻¹ is the speed of light, and λ is the wavelength of radiation (meters). Then we have

$$\begin{aligned} nQ &= 1000 \text{ W} = 1000 \text{ J s}^{-1} \\ \Rightarrow n &= \frac{1000}{Q} = \frac{1000}{hc/\lambda_{\text{average}}} = 2.8959 \times 10^{21} \end{aligned}$$

where $\lambda_{\text{average}} = 575$ nanometers is the average wavelength of the solar spectrum.

2. Imagine a room 10 by 20 meters with a ceiling 5 meters high. The room has no windows, and there is a light bulb shining at one corner of the ceiling. The light bulb has a radiant intensity of 10 W/sr. Consider a circular patch at the center of the floor with a 10 cm diameter.

a. What is the solid angle from the bulb to the patch?

Solution: The room and the solid angle from the corner to the tilted patch are shown in Figure 1.

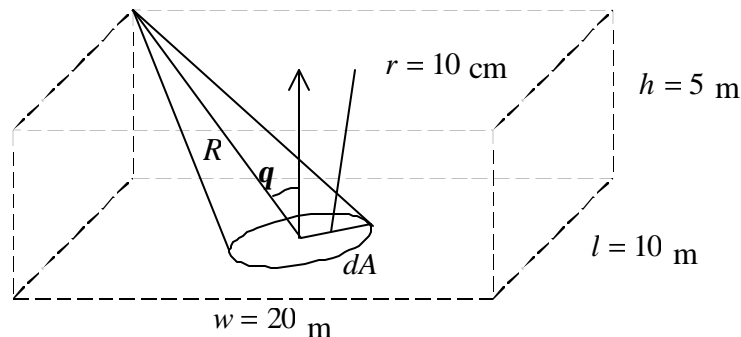


Figure 1: A room without windows

We have

$$dA = \mathbf{p} \cdot \mathbf{r}^2 = 0.1257 \text{ m}^2$$

$$R = \sqrt{h^2 + \left(\frac{\sqrt{w^2 + l^2}}{2} \right)^2} = 12.2474 \text{ m}$$

$$\cos \mathbf{q} = \frac{h}{R} = 0.4082$$

Therefore, the solid angle is approximately

$$d\Omega = \frac{dA \cdot \cos \theta}{R^2} = \frac{0.1257 \times 0.4082}{12.2474^2} = 3.4201 \times 10^{-4} \text{ sr}$$

- b. **What is the irradiance of the patch of floor? (Assume there is no reflection from the wall and from the ceiling.)**

Solution: Because radiant intensity is

$$I = 10 \text{ W/sr} ,$$

we have

$$dP = I * d\Omega = 0.0034 \text{ W} .$$

Considering that there is no reflection from the wall and from the ceiling, the irradiance of the patch of floor

$$E = dP/dA = 0.0272 \text{ W/m}^2$$

3. **We mix blue paint and yellow paint. Blue reflects only blue, and yellow absorbs all blue. What should be the color of the mixed paint? Typically, however, the mixed paint is green. Explain this.**

Solution: If the blue paint and the yellow paint were ideal, i.e. the blue paint only reflects blue and the yellow absorbs all blue, the mixed paint should be black. However, typically, the mixed paint is green. We can explain this as follow.

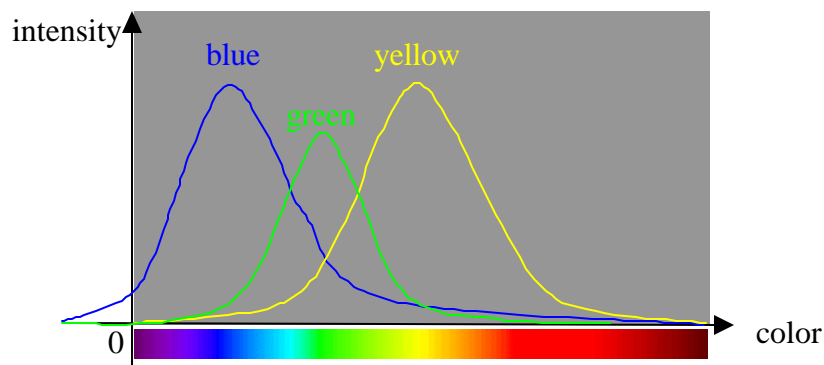


Figure 2: Mixed paint.

In practice, no paint is ideal. For example, the blue paint mainly reflects blue, but also reflects some other color near the blue. Because the blue is near the green on one side, the blue paint reflects some green though it mainly reflects blue. Yellow paint reflects green as well as red,

but the red is absorbed by the blue paint. So some green is reflected by both pigments, resulting in green paint.

4. One term we did not discuss is *albedo*. Explore this concept using the web or a textbook and provide a mathematical definition.

Solution: Generally speaking, the *albedo* is the ratio which the light reflected from an unpolished surface bears to the total light falling upon that surface, i.e. the fraction of incident electromagnetic radiation reflected by a surface, especially of a celestial body¹.

More precisely, in physics, the *albedo* is the ratio of loss due to scattering and loss due to total absorption. There are three basic effects where light passes through a medium — *absorption*, *emission*, and *scattering*. The *albedo* can be mathematically defined as:

$$\frac{\mathbf{s}(\mathbf{y}, \vec{\mathbf{w}}, \nu)}{\mathbf{c}(\mathbf{y}, \vec{\mathbf{w}}, \nu)} = \frac{\mathbf{s}(\mathbf{y}, \vec{\mathbf{w}}, \nu)}{\mathbf{k}(\mathbf{y}, \vec{\mathbf{w}}, \nu) + \mathbf{s}(\mathbf{y}, \vec{\mathbf{w}}, \nu)}$$

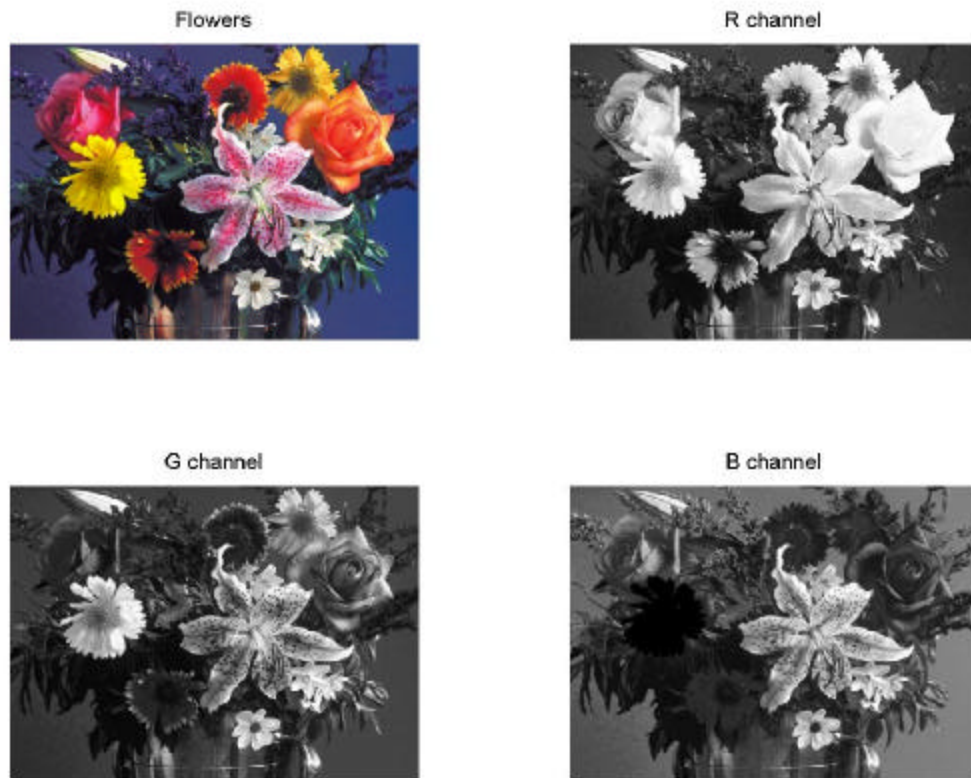
where $\mathbf{s}(\mathbf{y}, \vec{\mathbf{w}}, \nu)$ is the *scattering coefficient* describing the fraction of the incoming radiation scattered from its original direction, $\mathbf{k}(\mathbf{y}, \vec{\mathbf{w}}, \nu)$ is the *true absorption coefficient* describing the fraction of radiance lost per unit distance traveled through the medium, $\mathbf{c}(\mathbf{y}, \vec{\mathbf{w}}, \nu)$ is *total absorption coefficient* or the *extinction coefficient*, \mathbf{y} is the location of the point under calculation, $\vec{\mathbf{w}}$ is the incoming radiance direction, and ν is the incoming radiance frequency².

¹ Source: <http://www.dictionary.com>

² Source: <http://graphics.cs.uni-sb.de/~slusallek/Doc/html/node7.html>

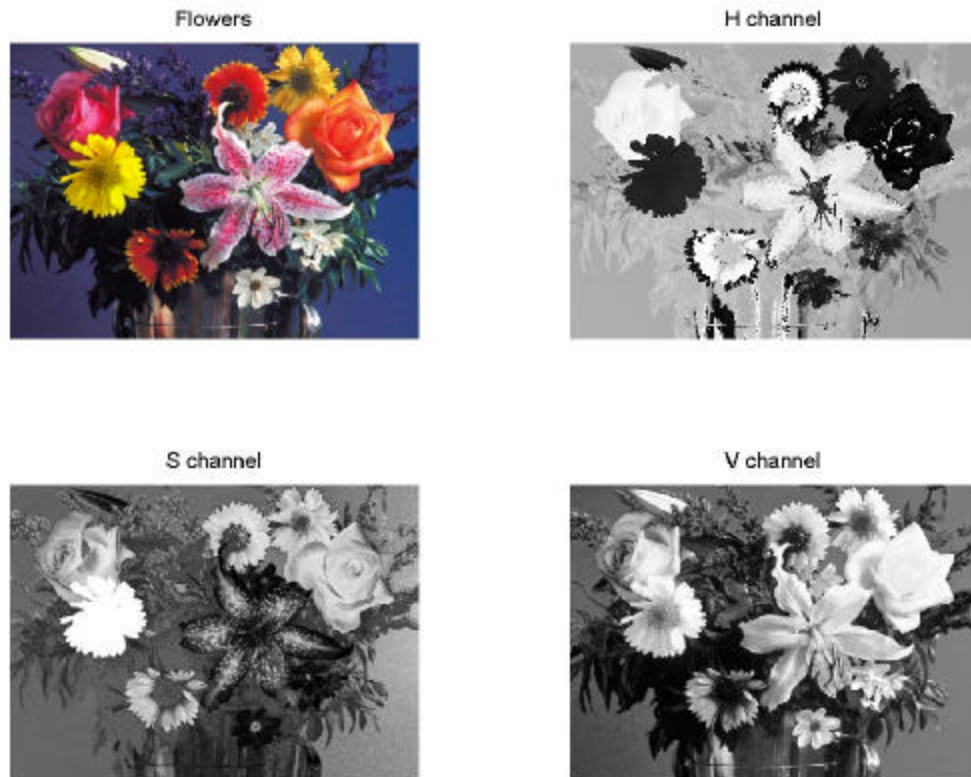
5. Read the RGB image called `flowers.tif` (available with all Matlab packages) using `imread('flowers.tif')`. Display the red, green and blue components of this image as 3 gray level images on the same figure (using `subplot`) and print the result.

Solution. The script is listed in Appendix.



6. Transform the flowers image to HSV color space using the Matlab function `rgb2hsv`. Display and print a single figure showing the H, S, V components of this image as 3 gray level images.

Solution. The script is listed in Appendix.

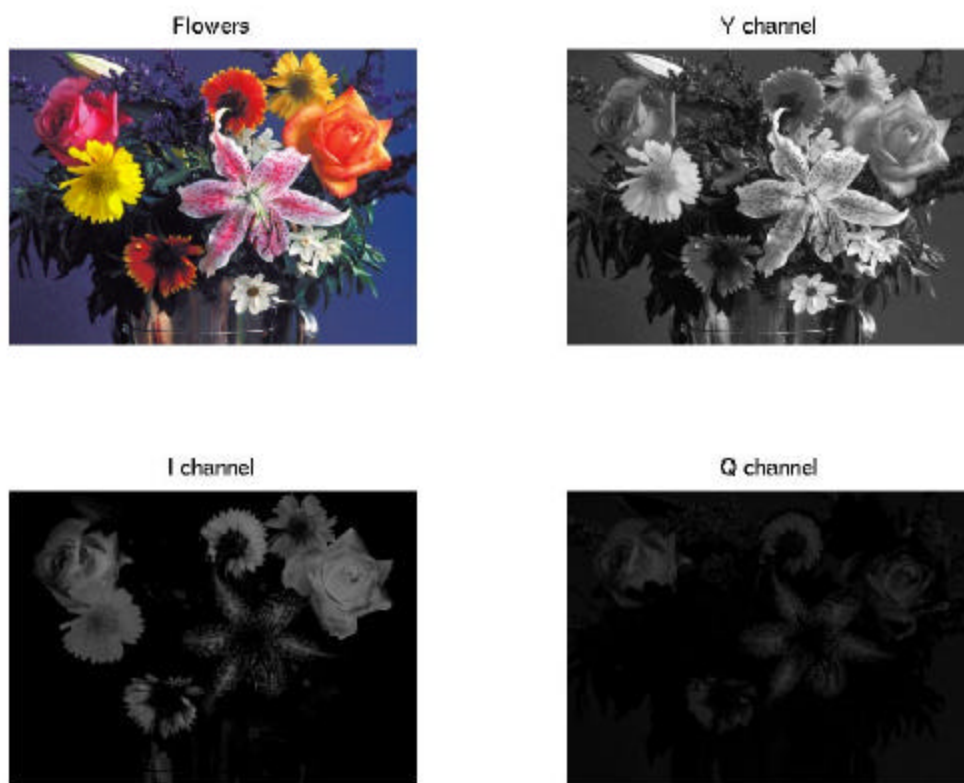


7. Transform the flowers image to YIQ color space using the transformation

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Again using `subplot`, display and print a single figure showing the Y, I, Q components of this image as 3 gray level images.

Solution. The script is listed in Appendix.



8. Transform the RGB flowers image into a gray level image using the function `rgb2gray`. Compare with the images constructed above. What method was used?

Solution:



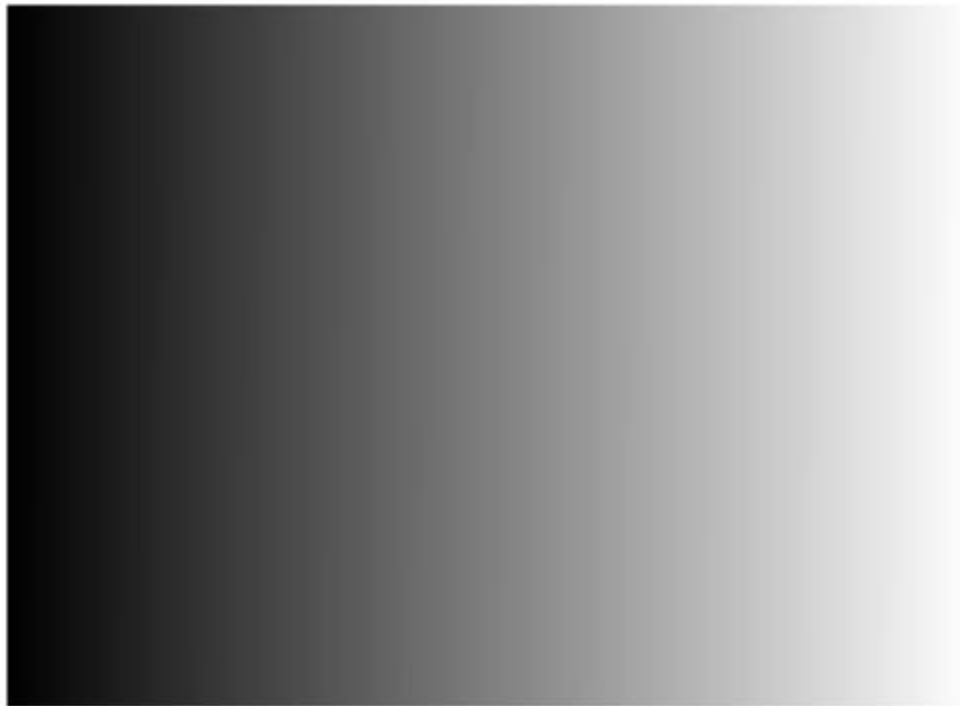
The YIQ color space transformation is used in `rgb2gray`. The gray level image is the Y channel of the YIQ space image. The script is listed in Appendix.

9. Look at the Matlab code for `rgb2ntsc` and `ntsc2rgb` by using the function `type`. What is the relationship between the matrix `T` used in the code and the matrix used in the YIQ transformation above?

Solution: The matrix `T` used in the code is the inverse matrix of the YIQ transformation above, i.e. $T * T_{YIQ} = I$.

10. Transform the RGB flowers image into an indexed image using the function `rgb2ind`. How many indexes are used? Display and print a figure showing the index part of the image as a gray level image, with indexes mapped to the range 0 to 255.

Solution: The results are shown below. When no number of indexes is specified as argument, a total of $362 \times 500 = 181000$ indexes are used, i.e. a new index is assigned to each pixel, from 0 for the top left pixel to 181000 for the bottom right pixel. These values are rescaled from 0 to 255 by a division by 255, or are rescaled automatically by using `imshow(flowers_index, [])`. The script is listed in Appendix.



Appendix: Scripts for question 5, 6, 7, 8, 10

```
%=====
%= HW2_5

% Read the image from file.
flowers_RGB = imread('flowers.tif');

% Get the RGB channel.
flowers_R = flowers_RGB(:, :, 1);
flowers_G = flowers_RGB(:, :, 2);
flowers_B = flowers_RGB(:, :, 3);

% Show results.
figure;

subplot(2, 2, 1);
imshow(flowers_RGB);
title('Flowers');

subplot(2, 2, 2);
imshow(flowers_R);
title('R channel');

subplot(2, 2, 3);
imshow(flowers_G);
title('G channel');

subplot(2, 2, 4);
imshow(flowers_B);
title('B channel');

% Save results into a file.
print -djpeg flowers_RGB;

%=====
%= HW2_6
```

```
% Convert the RGB image to HSV image.
flowers_HSV = rgb2hsv(flowers_RGB);

% Get the HSV channels.
flowers_H = flowers_HSV(:, :, 1);
flowers_S = flowers_HSV(:, :, 2);
flowers_V = flowers_HSV(:, :, 3);

% Show results.
figure;

subplot(2, 2, 1);
imshow(flowers_RGB);
title('Flowers');

subplot(2, 2, 2);
imshow(flowers_H);
title('H channel');

subplot(2, 2, 3);
imshow(flowers_S);
title('S channel');

subplot(2, 2, 4);
imshow(flowers_V);
title('V channel');

% Save results into a file.
print -djpeg flowers_HSV;

%=====
%= HW2_7

% Convert the RGB image to HSV image.
T_YIQ = [ ...
    0.299, 0.587, 0.114; ...
```

```
    0.596, -0.275, -0.321; ...
    0.212, -0.523  0.311 ];

nRow = size(flowers_RGB, 1);
nCol = size(flowers_RGB, 2);

rgb = reshape(double(flowers_RGB)/255, nRow * nCol, 3);
yiq = T_YIQ * double(rgb)/255;

flowers_YIQ = reshape(yiq', nRow, nCol, 3);

% Get the YIQ channels.
flowers_Y = flowers_YIQ(:, :, 1);
flowers_I = flowers_YIQ(:, :, 2);
flowers_Q = flowers_YIQ(:, :, 3);

% Show results.
figure;

subplot(2, 2, 1);
imshow(flowers_RGB);
title('Flowers');

subplot(2, 2, 2);
imshow(flowers_Y);
title('Y channel');

subplot(2, 2, 3);
imshow(flowers_I);
title('I channel');

subplot(2, 2, 4);
imshow(flowers_Q);
title('Q channel');

% Save results into a file.
print -djpeg flowers_YIQ;
```

```
%=====
%= HW2_8

% Convert the color image into a gray image.
flowers_gray = rgb2gray(flowers_RGB);

% show result.
figure

subplot(1, 2, 1);
imshow(flowers_RGB);
title('Flowers');

subplot(1, 2, 2);
imshow(flowers_gray);
title('Gray level image');

% Save results into a file.
print -djpeg flowers_gray;

%=====
%= HW2_10

% Convert the color image into an index image.
[flowers_index, map] = rgb2ind(flowers_RGB);

% show result.
figure
imshow(flowers_index, []);

% Save results into a file.
print -djpeg flowers_index;
```