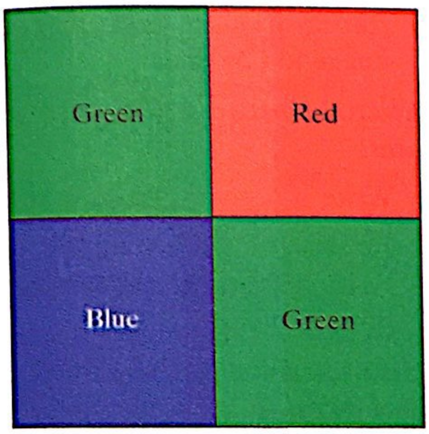
7.24* Given an image in the RGB, CMY, or CMYK color system, how would you implement the color equivalent of gray-scale histogram matching (specification) from Section 3.3?

7.25 Consider the following 500×500 RGB image, in which the squares are fully saturated red, green, and blue, and each of the colors is at maximum intensity. An HSI image is generated from this image. Answer the following questions.



(a) Describe the appearance of each HSI component image.

(b)* The saturation component of the HSI image is smoothed using an averaging kernel of size 125×125. Describe the appearance of the result. (You may ignore image border effects in the filtering operation.)

(c) Repeat (b) for the hue image.

7.26 Answer the following.

(a)* Refer to the discussion in Section 7.7 about segmentation in the RGB color space. Give a procedure (in flow chart form) for determining whether a color vector (point) $\mathbf{z}$ is inside a cube with sides W, centered at an average color vector $\mathbf{a}$. Distance computations are not allowed.

(b) If the box is aligned with the axes this process also can be implemented on an image-by-image basis. Show how you would do it.

7.27 Show that Eq. (7-49) reduces to Eq. (7-48) when $\mathbf{C} = \mathbf{I}$, the identity matrix.

7.28 Sketch the surface in RGB space for the points that satisfy the equation

$$D(\mathbf{z},\mathbf{a}) = \left[(\mathbf{z}-\mathbf{a})^T \mathbf{C}^{-1} (\mathbf{z}-\mathbf{a})\right]^{\frac{1}{2}} = D_0$$

where D_0 is a positive constant. Assume that $\mathbf{a} = \mathbf{0}$, and that

$$\mathbf{C} = \begin{bmatrix} 8 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

7.29 Refer to the discussion on color edge detection in Section 7.7. One might think that a logical approach for defining the gradient of an RGB image at any point (x, y) would be to compute the gradient vector (see Section 3.6) of each component image and then form a gradient vector for the color image by summing the three individual gradient vectors. Unfortunately, this method can at times yield erroneous results. Specifically, it is possible for a color image with clearly defined edges to have a zero gradient if this method were used. Give an example of such an image. (*Hint:* To simplify your analysis, set one of the color planes to a constant value.)

Projects

MATLAB solutions to the projects marked with an asterisk (*) are in the DIP4E Student Support Package (consult the book web site: www.ImageProcessingPlace.com).

7.1 RGB color cube.

(a)* (a) Write a function **g = rgbcube4e(vz,vy,vz)** to generate and display the RGB color cube in Fig. 7.8 (see Fig. 7.7 for axis-color definitions). The inputs are the three coordinates of your viewing position with reference to the origin of the cube. You should be able to view the cube from any 3-D viewpoint, and be able to extract any of its face images. Output **g** is an image of the cube displayed by this function. (*Hint:* Consider using MATLAB function **patch** to generate the cube, and the pair of functions **getframe** and **frame2im** to capture **g**.)