

Supplement to Computer Science Chapter 7 – Digital Video Processing

[Programming Exercise – Digital Video Processing > Chroma Keying¹](#)

Introduction:

Chroma keying is a method for removing certain colors from an image. It is the basis for bluescreening and greenscreening, which is used when you want to erase the background of an object, replacing it with a different one.

Chroma keying uses a multiplier, α , to specify the opacity level of a pixel or of a color component of a pixel.

One simple approach to chroma keying operates on colors in HLS color mode, as described below:

Let the color of a pixel be $C = (h, l, s)$ where $0 \leq h, l, s \leq 1$. Then given a hue h_{key} that is to be eliminated from an image and a tolerance t , the alpha value for the pixel is computed by

$$\alpha = 0 \text{ if } (h_{key} - t) < h < (h_{key} + t) \text{ and } \alpha = 1 \text{ otherwise}$$

Method 1

In this method, which we call Method 1, an unwanted hue h_{key} is removed by setting $\alpha = 0$ for all pixels of hue h that are within tolerance t of h_{key} .

Method 1 can be refined by computing α as a weighted average of alpha values determined for each of the three components, as follows:

¹This material is based on work supported by the National Science Foundation under Grant No. DUE-0127280. This programming assignment was written by Dr. Jennifer Burg (burg@wfu.edu).

Let the color of a pixel be $C = (h, l, s)$ where $0 \leq h, l, s \leq 1$. Let weights be given as w_h , w_l , and w_s , where $w_h + w_l + w_s = 1$. Then given a color $K = (h_{key}, l_{key}, s_{key})$ that is to be eliminated from an image and tolerances t_h , t_l , and t_s , the alpha value, α , for the pixel is computed by

$$\alpha = w_h \alpha_h + w_l \alpha_l + w_s \alpha_s$$

where

$\alpha_h = 0$ if $(h_{key} - t_h) < h < (h_{key} + t_h)$, and $\alpha_h = 1$

otherwise

$\alpha_l = 0$ if $(l_{key} - t_l) < l < (l_{key} + t_l)$, and $\alpha_l = 1$ otherwise

$\alpha_s = 0$ if $(s_{key} - t_s) < s < (s_{key} + t_s)$, and

$\alpha_s = 1$ otherwise

Method 2

A third method can be applied to RGB color. It is defined as follows:

Let the color of a pixel be $C = (c_r, c_g, c_b)$, where $0 \leq c_r, c_g, c_b \leq 1$.

Then the alpha value for the pixel is computed by the color difference keying method for bluescreening is

$$\alpha = 1 - (c_b - \max(c_r, c_g))$$

with the result clipped to the range [0 1].

Method 3

An analogous process can be done for greenscreening with

$$\alpha = 1 - (c_g - \max(c_r, c_b))$$

Instructions:

The Assignment

Using the programming language of your choice, implement the three methods for chroma keying given above. Your program should read in an image file, remove a green or blue background, and replace it with a different background. The new image should be written out to a file.

For Method 1, the program should take as input a hue h_{key} to be removed from images and a tolerance t .

For Method 2, the input parameters should be h_{key} , l_{key} , s_{key} , t_h , t_l , and t_s .

For Method 3, the input parameters should be a number indicating whether blue or greenscreening should be done.

All three programs should also take as input the name of the input and output files. You can prompt the user for input or put them on the command line.

For all three methods, your program should read a raw image file that has just three values per pixel representing the RGB components. For methods 1 and 2, the RGB components will have to be converted to HLS for chroma keying and then back to RGB before the image is written to a file.

Compare the results of the three methods.

Before Writing the Program

To run your program, you will need some raw RGB color image files as input, so create these first or ask your instructor if he or she plans to create them for you. Your program will need to know the width and height of the images it is processing. The input image files can be created by saving an image (BMP, JPG, etc.) in a raw format in a standard image processing program.