

☰ Documentation

[Accelerate](#) / [...](#) / [Transforming with lookup tables](#) / Applying transformations to selected colors in an image

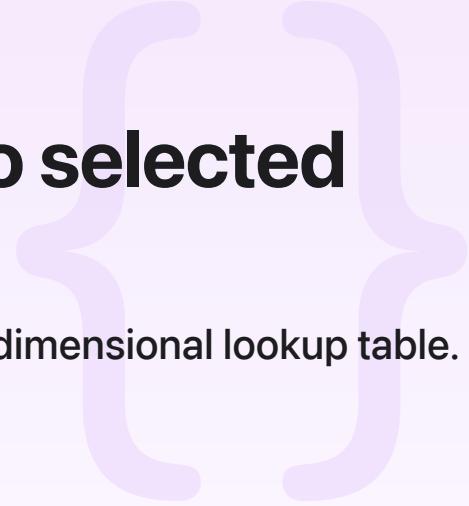
Sample Code

Applying transformations to selected colors in an image

Desaturate a range of colors in an image with a multidimensional lookup table.

[Download](#)

macOS 14.0+ | Xcode 15.2+



Overview

This sample code app allows you to desaturate the colors in an image based on their similarity to a selected color. For example, the image below shows an original image (left) and the transformed image (right) where the app has transformed the cake sprinkles that aren't yellow to grayscale.



Before exploring the code, try building and running the app to familiarize yourself with the effect of the different transformations on the image. In the app, click the smaller image to select a color and the app displays the transformed result as the larger image.

Create the lookup table data

The multidimensional lookup table that this sample code project uses is a 3D cube with entries PerChannel values along each axis.

The app defines the targetLabColor variable as the color that the user selects in the user interface.

```
let targetLabColor = ColorConverter.rgbToLab(red: targetRGB[0],  
                                             green: targetRGB.count > 1 ? targetRGB[1],  
                                              blue: targetRGB.count > 2 ? targetRGB[2])
```

The code below performs entriesPerChannel iterations from 0 to 1 over each color channel to populate the multidimensional lookup table data. With each iteration, the code calls simd_distance to compute the Euclidean distance in L*a*b* color space between the target color and the color that red, green, and blue loops define.

The distance between the two colors forms the basis for the amount of desaturation or darkening that the app applies to each color value in the lookup table.

```
var bufferIndex = 0  
let multiplier = CGFloat(UInt16.max)  
for red in ramp {  
    for green in ramp {  
        for blue in ramp {  
  
            let srcLabColor = ColorConverter.rgbToLab(red: red, green: green, blue:  
                blue)  
  
            let distance = simd_distance(targetLabColor, srcLabColor)  
  
            let effectMultiplier = 1 - simd_smoothstep(0, tolerance, distance)  
  
            let src = NSColor(red: red,  
                             green: green,  
                             blue: blue,  
                             alpha: 1)  
  
            let dest = NSColor(hue: src.hueComponent,  
                               saturation: src.saturationComponent * (desaturate ? 0 :  
                                brightness: src.brightnessComponent * (darken ? CGFl  
                                alpha: 1)  
  
            lookupTableData[ bufferIndex ] = UInt16(dest.redComponent * multiplier)
```

```

        bufferIndex += 1

        lookupTableData[ bufferIndex ] = UInt16(dest.greenComponent * multiplier)
        bufferIndex += 1

        lookupTableData[ bufferIndex ] = UInt16(dest.blueComponent * multiplier)
        bufferIndex += 1
    }
}
}

```

For more information about creating multidimensional lookup table data, see [Applying color transforms to images with a multidimensional lookup table](#).

Create the multidimensional lookup

The `vImage.MultidimensionalLookupTable` structure uses the lookup table to transform the source image, either desaturating or darkening colors that aren't close to the selected target color.

The code below creates a multidimension lookup table structure, applies the lookup table to the three source planar buffers, and writes the result to the three destination buffers:

```

let lookupTable = vImage.MultidimensionalLookupTable(
    entryCountPerSourceChannel: entryCountPerSourceChannel,
    destinationChannelCount: destChannelCount,
    data: lookupTableData)

lookupTable.apply(sources: rgbSourceBuffers,
                 destinations: [ redDestinationBuffer,
                                  greenDestinationBuffer,
                                  blueDestinationBuffer ],
                 interpolation: .full)

```

The three destination planar buffers contain the transformed image.

Create a Core Graphics image

The code below interleaves the three planar buffers into a single RGB buffer. Finally, the `makeCGImage(cgImageFormat:)` function returns a Core Graphics image that the app displays in the user interface.

```
interleavedDestinationBuffer.interleave(planarSourceBuffers: [ redDestinationBuffer,  
greenDestinationBuffer,  
blueDestinationBuffer])
```

See Also

Transforming with a multidimensional lookup table

- 📄 Applying color transforms to images with a multidimensional lookup table
Precompute translation values to optimize color space conversion and other pointwise operations.
- { } Cropping to the subject in a chroma-keyed image
Convert a chroma-key color to alpha values and trim transparent pixels using Accelerate.

```
func vImageMultidimensionalTable_Create(UnsafePointer<UInt16>, UInt32,  
UInt32, UnsafePointer<UInt8>, vImageMDTableUsageHint, vImage_Flags,  
UnsafeMutablePointer<vImage_Error>! ) -> vImage_MultidimensionalTable!
```

Creates a multidimensional lookup table.

```
func vImageMultiDimensionalInterpolatedLookupTable_PlanarF(Unsafe  
Pointer<vImage_Buffer>, UnsafePointer<vImage_Buffer>, UnsafeMutableRaw  
Pointer!, vImage_MultidimensionalTable, vImage_InterpolationMethod, v  
Image_Flags) -> vImage_Error
```

Uses a multidimensional lookup table to transform a 32-bit planar image.

```
func vImageMultiDimensionalInterpolatedLookupTable_Planar16Q12(Unsafe  
Pointer<vImage_Buffer>, UnsafePointer<vImage_Buffer>, UnsafeMutableRaw  
Pointer!, vImage_MultidimensionalTable, vImage_InterpolationMethod, v  
Image_Flags) -> vImage_Error
```

Uses a multidimensional lookup table to transform a 16Q12 planar image.

```
func vImageMultidimensionalTable_Retain(vImage_MultidimensionalTable!)  
-> vImage_Error
```

Retains a multidimensional table.

```
func vImageMultidimensionalTable_Release(vImage_MultidimensionalTable!)  
-> vImage_Error
```

Releases a multidimensional table.

```
typealias vImage_MultidimensionalTable
```

An opaque pointer that represents a multidimensional lookup table.

```
struct vImageMDTableUsageHint
```

Constants that indicate the use for a multidimensional lookup table.

```
struct vImage_InterpolationMethod
```

Constants that represent different interpolation methods.