

[Accelerate](#) / [vImage](#) / `vImage.MultidimensionalLookupTable`

Structure

`vImage.MultidimensionalLookupTable`

A multidimensional lookup table.

iOS 16.0+ | iPadOS 16.0+ | Mac Catalyst | macOS 13.0+ | tvOS 16.0+ | visionOS | watchOS 9.0+

```
struct MultidimensionalLookupTable
```

Mentioned in

 Applying color transforms to images with a multidimensional lookup table

Overview

Use a multidimensional lookup table to transform the colors in an image. The lookup table defines an output color based on the input color values. The `vImage` multidimensional lookup table provides interpolation to compute output color values that don't have an explicit entry in the table for a given input color.

A `vImage.MultidimensionalLookupTable` applies transforms to 32-bit planar pixel buffers. There are pixel buffer functions available to convert between bit depths and interleaved to planar buffers.

The following is an example of a simple lookup table that implements the Rec. 709 luma coefficients to convert from a 3-channel RGB image to a single-channel grayscale image. The lookup table is a 3D cube with 32 entries per channel.

```
let entriesPerChannel = UInt8(32)
let srcChannelCount = 3
let destChannelCount = 1
```

```

let lookupTableElementCount = Int(pow(Float(entriesPerChannel),
                                     Float(srcChannelCount))) *
Int(destChannelCount)

let tableData = [UInt16](unsafeUninitializedCapacity: lookupTableElementCount) {
    buffer, count in

    let multiplier = Float(UInt16.max)
    var bufferIndex = 0

    for red in ( 0 ..< entriesPerChannel) {
        for green in ( 0 ..< entriesPerChannel) {
            for blue in ( 0 ..< entriesPerChannel) {

                let normalizedRed = Float(red) / Float(entriesPerChannel - 1)
                let normalizedGreen = Float(green) / Float(entriesPerChannel - 1)
                let normalizedBlue = Float(blue) / Float(entriesPerChannel - 1)

                let gray = (normalizedRed * 0.2126) +
                           (normalizedGreen * 0.7152) +
                           (normalizedBlue * 0.0722)

                buffer[ bufferIndex ] = UInt16(gray * multiplier)
                bufferIndex += 1
            }
        }
    }

    count = lookupTableElementCount
}

```

Use the lookup table data to create a [vImage.MultidimensionalLookupTable](#) structure.

```

let entryCountPerSourceChannel = [UInt8](repeating: entriesPerChannel,
                                           count: srcChannelCount)

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

```

Call the `lookupTable.apply(sources:destinations:interpolation:)` function to transform a 3-channel RGB image to a grayscale image. In this example, the source is interleaved. The code demonstrates calling `planarBuffers()` to deinterleave the source.

```
let src = vImage.PixelBuffer<vImage.InterleavedFx3>( ... )

let planarSources = src.planarBuffers()

let dest = vImage.PixelBuffer<vImage.PlanarF>(size: src.size)

lookupTable.apply(sources: planarSources,
                  destinations: [dest],
                  interpolation: .none)
```

On return, `dest` contains a grayscale representation of the source RGB image.

Topics

Initializers

`init<T>(entryCountPerSourceChannel: [UInt8], destinationChannelCount: Int, data: T)`

Returns a new multidimensional lookup table.

Instance Properties

`let destinationChannelCount: Int`

The number of destination channels.

`let entryCountPerSourceChannel: [UInt8]`

An array that contains the number of table entries for each dimension of the lookup table.

`let sourceChannelCount: Int`

The number of source channels.

Instance Methods

```
func apply<SrcFormat, DestFormat>(source: vImage.PixelBuffer<SrcFormat>, destination: vImage.PixelBuffer<DestFormat>, interpolation: vImage.MultidimensionalLookupTable.InterpolationMethod)
```

Transforms a multiple plane pixel buffer using the multidimensional lookup table.

```
func apply(sources: [vImage.PixelBuffer<vImage.PlanarF>], destinations: [vImage.PixelBuffer<vImage.PlanarF>], interpolation: vImage.MultidimensionalLookupTable.InterpolationMethod)
```

Transforms an array of planar pixel buffers using the multidimensional lookup table.

Enumerations

```
enum InterpolationMethod
```

Describes the method a multidimensional lookup table uses to generate interpolated values between lookup table values.

See Also

Related Documentation



Applying color transforms to images with a multidimensional lookup table

Precompute translation values to optimize color space conversion and other pointwise operations.

Type Aliases

```
 typealias StructuringElement
```

A 2D matrix that represents a morphology kernel.

```
 struct ConvolutionKernel
```

Constants that describe 1D convolution kernels.

```
 struct ConvolutionKernel2D
```

A 2D matrix that represents a convolution kernel.

```
 struct DynamicPixelFormat
```

A buffer that contains pixels with a data type that's unknown at compile time.

`struct Interleaved16Fx2`

A two-channel, 16-bit-per-channel, floating-point interleaved buffer.

`struct Interleaved16Fx4`

A four-channel, 16-bit-per-channel, floating-point interleaved buffer.

`struct Interleaved16Ux2`

A two-channel, 16-bit-per-channel, unsigned-integer interleaved buffer.

`struct Interleaved16Ux4`

A four-channel, 16-bit-per-channel, unsigned-integer interleaved buffer.

`struct Interleaved8x2`

A two-channel, 8-bit-per-channel interleaved buffer.

`struct Interleaved8x3`

A three-channel, 8-bit-per-channel interleaved buffer.

`struct Interleaved8x4`

A four-channel, 8-bit-per-channel interleaved buffer.

`struct InterleavedFx2`

A two-channel, 32-bit-per-channel, floating-point interleaved buffer.

`struct InterleavedFx3`

A three-channel, 32-bit-per-channel, floating-point interleaved buffer.

`struct InterleavedFx4`

A four-channel, 32-bit-per-channel, floating-point interleaved buffer.

`struct Options`

Set flags on vImage operations to specify processing options.