Documentation Language: Swift

Accelerate / vImageHistogramCalculation_ARGB8888(_:_:_:)

Function

vImageHistogramCalculation_ARGB8888(_:_:_:)

Calculates the histogram of an 8-bit-per-channel, 4-channel interleaved buffer.

iOS 5.0+ | iPadOS 5.0+ | Mac Catalyst 13.1+ | macOS 10.3+ | tvOS 5.0+ | visionOS 1.0+ | watchOS 1.0+

```swift
func vImageHistogramCalculation_ARGB8888(
    _ src: UnsafePointer<vImage_Buffer>,
    _ histogram: UnsafeMutablePointer<UnsafeMutablePointer<vImagePixelCount>?>,
    _ flags: vImage_Flags
) -> vImage_Error
```

Parameters

`src`

The source vImage buffer.

`histogram`

An array of four collections that contain 256 elements that receive the histogram data.

`flags`

The options to use when performing the operation. If your code implements its own tiling or its own multithreading, pass `kvImageDoNotTile`; otherwise, pass `kvImageNoFlags`.

To specify that the function doesn’t calculate the alpha channel histogram, set the `kvImageLeaveAlphaUnchanged` flag.

Return Value

kvImageNoError; otherwise, one of the error codes in Data Types and Constants.

Discussion

The following code populates the histogramAlpha , histogramRed, histogramGreen, and histogramBlue arrays with the histograms for each channel of the specified vImage_Buffer structure.

```
var histogramAlpha = [vImagePixelCount](repeating: 0, count: 256)
var histogramRed = [vImagePixelCount](repeating: 0, count: 256)
var histogramGreen = [vImagePixelCount](repeating: 0, count: 256)
var histogramBlue = [vImagePixelCount](repeating: 0, count: 256)

histogramAlpha.withUnsafeMutableBufferPointer { zeroPtr in
    histogramRed.withUnsafeMutableBufferPointer { onePtr in
        histogramGreen.withUnsafeMutableBufferPointer { twoPtr in
            histogramBlue.withUnsafeMutableBufferPointer { threePtr in

                var histogramBins = [zeroPtr.baseAddress, onePtr.baseAddress,
                                     twoPtr.baseAddress, threePtr.baseAddress]

                histogramBins.withUnsafeMutableBufferPointer { histogramBinsPtr in
                    // `buffer` is a `vImage_Buffer` structure.
                    _ = vImageHistogramCalculation_ARGB8888(&buffer,
                                                            histogramBinsPtr.baseAdd
                                                            vImage_Flags(kvImageNoFl
                }
            }
        }
    }
}
```

See Also

Related Documentation

Enhancing image contrast with histogram manipulation

Enhance and adjust the contrast of an image with histogram equalization and contrast stretching.

{} Specifying histograms with vImage

Calculate the histogram of one image, and apply it to a second image.

Calculating a histogram

```
func vImageHistogramCalculation_Planar8(UnsafePointer<vImage_Buffer>, UnsafeMutablePointer<vImagePixelCount>, vImage_Flags) -> vImage_Error
```

Calculates the histogram of an 8-bit planar buffer.

```
func vImageHistogramCalculation_PlanarF(UnsafePointer<vImage_Buffer>, UnsafeMutablePointer<vImagePixelCount>, UInt32, Pixel_F, Pixel_F, vImage_Flags) -> vImage_Error
```

Calculates the histogram of a 32-bit planar buffer.

```
func vImageHistogramCalculation_ARGBFFFF(UnsafePointer<vImage_Buffer>, UnsafeMutablePointer<UnsafeMutablePointer<vImagePixelCount>?>, UInt32, Pixel_F, Pixel_F, vImage_Flags) -> vImage_Error
```

Calculates the histogram of a 32-bit-per-channel, 4-channel interleaved buffer.