Accelerate / vImage

API Collection

vImage

Manipulate large images using the CPU’s vector processor.

Overview

vImage is a high-performance image processing framework. It includes functions for image manipulation—convolutions, geometric transformations, histogram operations, morphological transformations, and alpha compositing—as well as utility functions for format conversions and other operations.

vImage optimizes image processing by using the CPU’s vector processor. If a vector processor is not available, vImage uses the next best available option. This framework allows you to reap the benefits of vector processors without the need to write vectorized code.

vImage is particularly suited for:

- Efficiently processing large images
- Real-time video processing software
- Scientific applications that require high-accuracy numerical calculations

Topics

First Steps

Converting bitmap data between Core Graphics images and vImage buffers

Pass image data between Core Graphics and vImage to create and manipulate images.

Creating and Populating Buffers from Core Graphics Images

Initialize vImage buffers from Core Graphics images.

📄 Creating a Core Graphics Image from a vImage Buffer

Create displayable representations of vImage buffers.

📄 Building a Basic Image-Processing Workflow

Resize an image with vImage.

📄 Compositing images with alpha blending

Combine two images by using alpha blending to create a single output.

📄 Compositing images with vImage blend modes

Combine two images by using blend modes to create a single output.

📄 Applying geometric transforms to images

Reflect, shear, rotate, and scale image buffers using vImage.

📄 Applying vImage operations to regions of interest

Limit the effect of vImage operations to rectangular regions of interest.

Swift Overlay

```
enum vImage
```

An enumeration that acts as a namespace for Swift overlays to vImage.

vImage Pixel Buffers

{} Using vImage pixel buffers to generate video effects

Render real-time video effects with the vImage Pixel Buffer.

{} Applying tone curve adjustments to images

Use the vImage library's polynomial transform to apply tone curve adjustments to images.

{} Adjusting the brightness and contrast of an image

Use a gamma function to apply a linear or exponential curve.

{} Adjusting the hue of an image

Convert an image to L*a*b* color space and apply hue adjustment.

{} Sharing texture data between the Model I/O framework and the vImage library

Use Model I/O and vImage to composite a photograph over a computer-generated sky.

{} Calculating the dominant colors in an image

Find the main colors in an image by implementing k-means clustering using the Accelerate framework.

`struct PixelBuffer`

An image buffer that stores an image’s pixel data, dimensions, bit depth, and number of channels.

vImage Buffers

Optimizing image-processing performance

Improve your app’s performance by converting image buffer formats from interleaved to planar.

vImage buffers

Use buffers to pass image data to and from vImage operations.

Core Graphics Interoperability

Core Graphics interoperability

Pass image data between the Core Graphics framework and the vImage library.

Core Video Interoperability

{} Integrating vImage pixel buffers into a Core Image workflow

Share image data between Core Video pixel buffers and vImage buffers to integrate vImage operations into a Core Image workflow.

{} Applying vImage operations to video sample buffers

Use the vImage convert-any-to-any functionality to perform real-time image processing of video frames streamed from your device’s camera.

{} Converting luminance and chrominance planes to an ARGB image

Create a displayable ARGB image using the luminance and chrominance information from your device’s camera.

{} Improving the quality of quantized images with dithering

Apply dithering to simulate colors that are unavailable in reduced bit depths.

Core Video interoperability

Pass image data between Core Video and vImage.

vImage Operations

Adjusting saturation and applying tone mapping

Convert an RGB image to discrete luminance and chrominance channels, and apply color and contrast treatments.

Blurring an image

Filter an image by convolving it with custom and high-speed kernels.

Adding a bokeh effect to images

Simulate a bokeh effect by applying dilation.

Converting color images to grayscale

Convert an RGB image to grayscale using matrix multiplication.

Building a basic image conversion workflow

Learn the fundamentals of the convert-any-to-any function by converting a CMYK image to an RGB image.

Specifying histograms with vImage

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

Enhancing image contrast with histogram manipulation

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

Reducing artifacts with custom resampling filters

Implement custom linear interpolation to prevent the ringing effects associated with scaling an image with the default Lanczos algorithm.

Finding the sharpest image in a sequence of captured images

Share image data between vDSP and vImage to compute the sharpest image from a bracketed photo sequence.

vImage Operations

Apply image manipulation operations to vImage buffers.

Data Types and Constants

Data Types and Constants

Look up type aliases, data types, and constants the vImage library uses.

Macros

vImage Macros

See Also

Image Processing Essentials

Converting bitmap data between Core Graphics images and vImage buffers

Pass image data between Core Graphics and vImage to create and manipulate images.

Creating and Populating Buffers from Core Graphics Images

Initialize vImage buffers from Core Graphics images.

Creating a Core Graphics Image from a vImage Buffer

Create displayable representations of vImage buffers.

Building a Basic Image-Processing Workflow

Resize an image with vImage.

Applying geometric transforms to images

Reflect, shear, rotate, and scale image buffers using vImage.

Compositing images with alpha blending

Combine two images by using alpha blending to create a single output.

Compositing images with vImage blend modes

Combine two images by using blend modes to create a single output.

Applying vImage operations to regions of interest

Limit the effect of vImage operations to rectangular regions of interest.

Optimizing image-processing performance

Improve your app’s performance by converting image buffer formats from interleaved to planar.