Function

vImageConvert_ARGBFFFFtoARGB8888_dithered(_:_:_:_:_:_:_:)

Converts a floating-point 32-bit-per-channel, 4-channel buffer to an 8-bit-per-channel, 4-channel buffer using the specified dithering algorithm.

iOS 8.0+ | iPadOS 8.0+ | Mac Catalyst 13.1+ | macOS 10.10+ | tvOS 8.0+ | visionOS 1.0+ | watchOS 1.0+

```
func vImageConvert_ARGBFFFFtoARGB8888_dithered(
    _ src: UnsafePointer<vImage_Buffer>,
    _ dest: UnsafePointer<vImage_Buffer>,
    _ maxFloat: UnsafePointer<Float>,
    _ minFloat: UnsafePointer<Float>,
    _ dither: Int32,
    _ permuteMap: UnsafePointer<UInt8>!,
    _ flags: vImage_Flags
) -> vImage_Error
```

Parameters

`src`
A pointer to a vImage buffer structure that contains the source image whose data you want to convert.

`dest`
A pointer to the destination vImage buffer structure. You're responsible for filling out the `height`, `width`, and `rowBytes` fields of this structure, and for allocating a data buffer of the appropriate size. On return, the data buffer this structure points to contains the destination image data. When you no longer need the data buffer, deallocate the memory to prevent memory leaks.

`maxFloat`
The four maximum source pixel values.

`minFloat`
The four minimum source pixel values.

`dither`
Type of dithering to apply to the image, if any.

`permuteMap`
An array of four 8-bit integers with the values `0`, `1`, `2`, and `3`, in some order. Each value specifies the channel from the source image that the function copies to the destination channel at the corresponding index.

`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`.

Return Value

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

Discussion

This function supports the following dithering algorithms:

`kvImageConvert_DitherNone`
Doesn't apply any dithering. This algorithm rounds the input values to the nearest representable value in the destination format.

`kvImageConvert_DitherOrdered`
Adds precomputed blue noise to the source image before it rounds the input values to the nearest representable value in the destination format. The vImage conversion functions support uniform and Gaussian noise by including `kvImageConvert_OrderedUniformBlue` and `kvImageConvert_OrderedGaussianBlue`, respectively.

`kvImageConvert_DitherOrderedReproducible`
Returns the same result as `kvImageConvert_DitherOrdered` but uses the same offset into the blue noise for each call.

`kvImageConvert_DitherFloydSteinberg`
Applies Floyd-Steinberg dithering to the image.

`kvImageConvert_DitherAtkinson`
: Applies Atkinson dithering to the image.

See Also

Related Documentation

{} Improving the quality of quantized images with dithering

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

Converting from floating-point 32-bit-per-channel buffers

```
func vImageConvert_RGBFFFtoRGB888_dithered(UnsafePointer<vImage_Buffer
>, UnsafePointer<vImage_Buffer>, UnsafePointer<Pixel_F>, UnsafePointer<
Pixel_F>, Int32, vImage_Flags) -> vImage_Error
```

Converts a floating-point 32-bit-per-channel, 3-channel buffer to an 8-bit-per-channel, 3-channel buffer using the specified dithering algorithm.