

[Accelerate](#) / vImageRotate90_PlanarF(_:_:_:_:_)

Function

vImageRotate90_PlanarF(_:_:_:_:_:_)

Rotates a 32-bit planar image by a multiple of 90°.

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

```
func vImageRotate90_PlanarF(
    _ src: UnsafePointer<vImage_Buffer>,
    _ dest: UnsafePointer<vImage_Buffer>,
    _ rotationConstant: UInt8,
    _ backColor: Pixel_F,
    _ flags: vImage_Flags
) -> vImage_Error
```

Parameters

src

A pointer to a vImage buffer structure that contains the source image.

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.

rotationConstant

A constant that specifies the rotation angle as a multiple of 90°. See [Rotation constants](#) for the full list of supported rotation constants.

backColor

A background color. If you set the `kvImageBackgroundColorFill` flag, pass a pixel value.

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

`kvImage.NoError`; otherwise, one of the error codes from [Error codes](#).

Discussion

This function maps the center point of the source image to the center point of the destination image. It doesn't scale or resample; instead, the function copies unchanged individual pixels to new locations.

Depending on the relative sizes of the source image and the destination buffer, the function may clip parts of the source image. Areas outside the source image might appear in the destination image if you don't pass a background color to the function.

The 90° and 270° rotations don't rotate around the true center of the image if either of the following is true:

- The parities of the source height and the destination width don't match. For example, the source height is odd and the destination width is even.
- The parities of the source width and the destination height don't match. For example, the source width is odd and the destination height is even.

The 0° and 180° rotations don't rotate around the true center of the image if either of the following is true:

- The parities of the source height and the destination height don't match. For example, the source height is odd and the destination height is even.
- The parities of the source width and the destination width don't match. For example, the source width is odd and the destination width is even.

To overcome this limitation, use the high-level rotation functions — for example, `vImageRotate90_ARGB16U(: : : :)`.

See Also

Related Documentation

- 📄 Applying geometric transforms to images

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

Rotating 32-bit-per-channel buffers by multiples of 90°

```
func vImageRotate90_ARGBFFFF(UnsafePointer<vImage_Buffer>, Unsafe  
Pointer<vImage_Buffer>, UInt8, UnsafePointer<Float>, vImage_Flags) -> v  
Image_Error
```

Rotates a 32-bit-per-channel, 4-channel interleaved image by a multiple of 90°.