

[Accelerate](#) / Compositing images with vImage blend modes

Article

Compositing images with vImage blend modes

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

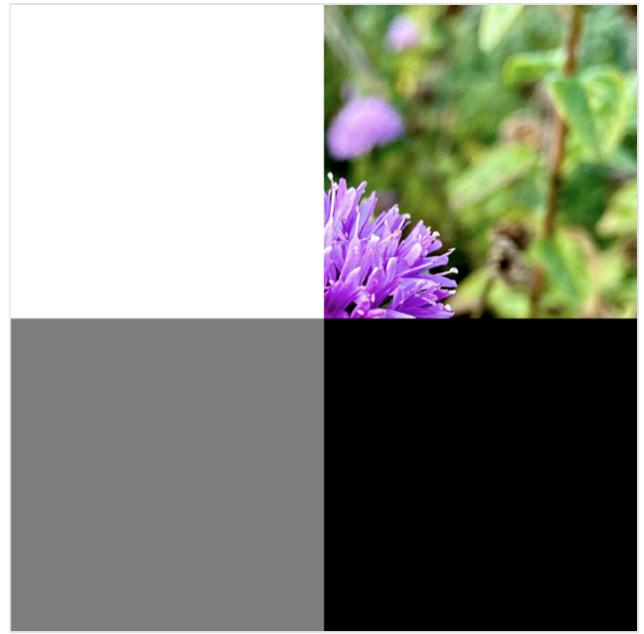


Overview

vImage provides a suite of functions for compositing two source images into one output. These *blend mode* functions implement different algorithms to determine the output color. For example, the multiply blend mode sets each output pixel as the product of the two corresponding input pixels.

The blend mode functions work on 8-bit RGBA source images in a premultiplied format. You can convert a nonpremultiplied buffer to a premultiplied format by using [vImagePremultiplyData_RGBA8888\(: : : \).](#)

The blend mode functions accept two [vImage_Buffer](#) structures as the bottom and top layers of the compositing operation. The examples that follow use the image below on the left as the bottom layer and the image below on the right as the top layer.



The top-layer source image consists of four sections:

- The top-left quadrant is solid white ([255, 255, 255]).
- The top-right quadrant is identical to the corresponding quadrant of the bottom image.
- The bottom-left quadrant is solid mid-gray ([127, 127, 127]).
- The bottom-right quadrant is solid black ([0, 0, 0]).

Both images are fully opaque.

Composite images using the lighten blend mode

Use the `vImagePremultipliedAlphaBlendLighen_RGBA8888(: : : :)` function to composite two layers using the lighten blend mode. You can use the lighten blend mode to apply an overall lightening effect to images.

The lighten blend mode sets each channel of the destination pixel as the lightest value for the corresponding channel of the two source layers. The image below shows the result of compositing using the lighten blend mode:



The top-left quadrant in the result is white because no pixels in the bottom layer are brighter than the corresponding pixels in the top layer.

The bottom-left quadrant in the result looks washed out because the operation selects gray pixels from the top layer over corresponding dark pixels from the bottom layer.

Composite images using the darken blend mode

Use the `vImagePremultipliedAlphaBlendDarken_RGBA8888(: : : :)` function to composite two layers using the darken blend mode. You can use the darken blend mode to apply an overall darkening effect to images.

The darken blend mode sets each channel of the destination pixel as the darkest value for the corresponding channel of the two source layers. The image below shows the result of compositing using the darken blend mode:



The bottom-right quadrant in the result is black because no pixels in the bottom layer are darker than the corresponding pixels in the top layer.

The bottom-left quadrant in the result looks darker than the corresponding quadrant in the bottom layer. This is because the operation selects gray pixels from the top layer over corresponding light pixels from the bottom layer.

Composite images using the screen blend mode

Use the `vImagePremultipliedAlphaBlendScreen_RGBA8888(: : : :)` function to composite two layers using the screen blend mode. You can use the screen blend mode to lighten images without affecting the darkest areas.

The screen blend mode sets the destination pixel as the inverted product of the inverted corresponding source pixels. The image below shows the result of compositing using the screen blend mode:



The bottom-right quadrant in the result is identical to the corresponding quadrant in the bottom layer because the operation multiplies each bottom-layer pixel value by 1.0. For example, if the source pixel value is 0.5, the destination pixel value is 0.5.

```
dest = 1 - (1 - 0.5) * (1 - 0.0) // dest = 0.5
```

The top-right quadrant in the result is brighter than the corresponding quadrant in the bottom layer. In this quadrant, the top-layer and bottom-layer pixel values are identical. For example, if the source pixel value is 0.5, the destination pixel value is 0.75.

```
dest = 1 - (1 - 0.5) * (1 - 0.5) // dest = 0.75
```

Composite images using the multiply blend mode

Use the `vImagePremultipliedAlphaBlendMultiply_RGBA8888(: : : :)` function to composite two layers using the multiply blend mode. You can use multiply blend mode to darken images without affecting the lightest areas.

The multiply blend mode sets the destination pixel as the product of the corresponding source pixels. The image below shows the result of compositing using the multiply blend mode:



The bottom-right quadrant in the result is black because the operation multiplies each bottom-layer pixel value by `0.0` from the top layer.

The top-left quadrant in the result is identical to the corresponding quadrant in the bottom layer because the operation multiplies each bottom-layer pixel value by `1.0`.

The top-right quadrant in the result is darker than the corresponding quadrant in the bottom layer. In this quadrant, the top-layer and bottom-layer pixel values are identical. For example, if the source pixel value is `0.5`, the destination pixel value is `0.25`.

```
dest = 0.5 * 0.5 // dest = 0.25
```

See Also

Related Documentation

☰ Alpha compositing

Composite images together.

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.
- 📄 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.
- ☰ vImage
Manipulate large images using the CPU's vector processor.