

ImageComposite

To generate an ROI masked filled with 1 in a region defined by a seed value and the boundary curves:

```
ImageBoundaryToMask  
ywave=yyy, xwave=xxx, width=100, height=200, scalingwave=src, seedx=550, seedy=700
```

See Also

The **ImageAnalyzeParticles** and **ImageSeedFill** operations. For another example see **Converting Boundary to a Mask** on page III-378.

ImageComposite

ImageComposite [/Z /FREE /DEST=destWave] srcImageA, srcImageB

The ImageComposite operation creates a new image by combining *srcImageA* and *srcImageB* subject to one of 12 Porter-Duff compositing modes.

The ImageComposite operation was added in Igor Pro 8.00.

Flags

/AALP=aWave	Specifies a 2D single-precision wave as the alpha associated with <i>srcImageA</i> . Alpha values are in the range 0 (transparent) to 1 (opaque).
/ACON=a1	Specifies a single alpha value for the whole <i>srcImageA</i> . <i>a1</i> is in the range 0 (transparent) to 1 (opaque).
/BALP=aWave	Specifies a 2D single-precision wave as the alpha associated with <i>srcImageB</i> . Alpha values are in the range 0 (transparent) to 1 (opaque).
/BCON=a1	Specifies a single alpha value for the whole <i>srcImageB</i> . <i>a1</i> is in the range 0 (transparent) to 1 (opaque).
/DEST=destWave	Specifies the wave to hold the composite image. If you omit /DEST the operation stores the image in the wave M_ImageComposite in the current data folder.
/FREE	Creates output wave as free waves. /FREE is permitted in user-defined functions only, not from the command line or in macros. If you use /FREE then destWave must be simple name, not a path.
/NMOD=mode	Selects one of the 12 Porter-Duff compositing modes. <i>mode</i> is a value from 1 to 12. The default is <i>mode</i> =4 corresponding to "A over B". See Compositing Modes on page V-368.
/OUT=layers	Specifies the number of layers of the output image. Valid values are 3 (RGB) or 4 (RGBA). By default the operation creates an RGB image.
/PMA=pmState	Set <i>pmState</i> to 1 if the RGB components in <i>srcImageA</i> are pre-multiplied. Use <i>pmState</i> =0 otherwise. By default <i>srcImageA</i> is assumed to be pre-multiplied. See Pre-multiplication on page V-368.
/PMB=pmState	Set <i>pmState</i> to 1 if the RGB components in <i>srcImageB</i> are pre-multiplied. Use <i>pmState</i> =0 otherwise. By default <i>srcImageB</i> is assumed to be pre-multiplied. See Pre-multiplication on page V-368.
/Z	No error reporting. The operation sets V_Flag to 0 if it succeeds or to an error code otherwise. You can use GetErrMessage to obtain a description of the error.

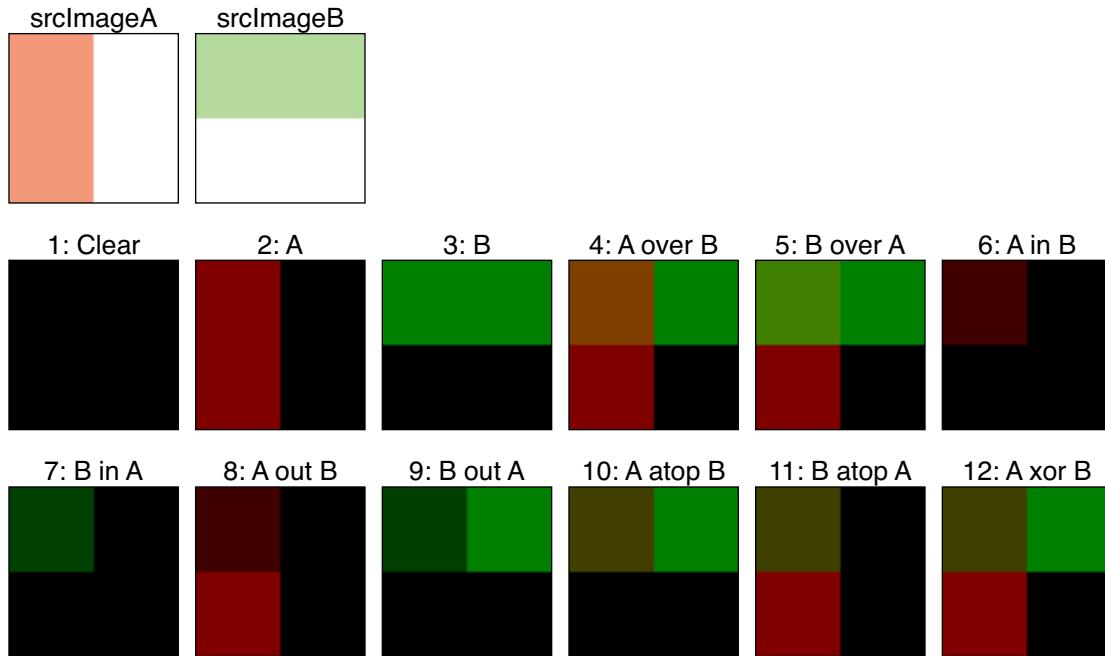
Pre-multiplication

An RGB value can be raw or pre-multiplied. "Pre-multiplied" means that the red, green, and blue values have been multiplied by normalized alpha values in the range 0 (transparent) to 1 (opaque).

ImageComposite operation is faster when working with pre-multiplied values. ImageComposite assumes that your RGB values are pre-multiplied unless you specify otherwise using /PMA=0 and /PMB=0.

Compositing Modes

Here are the 12 compositing modes supported by ImageComposite:



Details

ImageComposite computes an output RGB or RGBA image that result from compositing *srcImageA* and *srcImageB* using one of the Porter-Duff compositing modes shown in the table above. The waves *srcImageA* and *srcImageB* must have the same number of pixels and the same number type.

Supported number types are: unsigned char, unsigned short, unsigned int, single precision floating point and double precision floating point. When using integer waves expected alpha values are in the range [0,2^N-1] where N is the number of bits of the number type. Floating point waves should include alpha in the range [0,1].

There are three options to specify the alpha associated with each image:

1. The alpha can be expressed as the 4th layer in the wave.
2. The alpha can be specified by a single-precision wave that has the same number of pixels as the image using the /AALP and /BALP flags.
3. The alpha can be specified by a single number in the range [0,1] using the /ACON and /BCON flags.

Options 2 and 3 cannot override an alpha channel that is present in a source wave. To use these options you must delete the alpha channel in the source wave, if any.

Example

```

Function SetupImageCompositeDemo()           // Setup - Create two sample images
    Make/O/N=(128,128,4)/B/U imageA=0, imageB=0
    imageA[0,64][][0]=255
    imageA[0,64][][3]=128
    NewImage/S=0/N=imageAW imageA
    imageB[][],[0,64][1]=255
    imageB[],[0,64][3]=128
    NewImage/S=0/N=imageBW imageB
    AutoPositionWindow/M=0/R=imageAW imageBW
End

Function CompositeAOverB()                 // Composite A over B
    Wave imageA, imageB
    ImageComposite/PMA=0/PMB=0/DEST=M_Comp1/NMOD=4 imageA,imageB
    NewImage/S=0/N=comp1 M_Comp1
    AutoPositionWindow/M=0/R=imageBW
End

Function CompositeAInB()                  // Composite A in B
    Wave imageA, imageB
    ImageComposite/PMA=0/PMB=0/DEST=M_Comp2/NMOD=6 imageA,imageB
    NewImage/S=0/N=comp2 M_Comp2

```