Not sure this is exactly what you're asking for, but it is along the same lines. I typically deal with an arbitrary number of greyscale images which I want to combine into a multicolor image. What I do is render each greyscale image as an RGB image with a certain colormap. Once you have a set of RGB images, they can simply be added together to produce the final image.

Something like the code below. Note that each greyscale image is contained in an IDLgrImage object which have all been added to the same IDLgrView, and each has its own IDLgrPalette which sets the color of each channel. All of the images are hidden, and what I do is unhide them one at a time, saving an RGB image each time. At the end the RGB images are added up to form the final image.

=====================

```
output_image = intarr(3,xpix,ypix)

for i = 0, number_of_input_images-1 do begin

    image_object_array[i] -> SetProperty, HIDE=0
    image_buffer_object -> Draw, view_object
    image_buffer_object -> GetProperty, image_data = rgb_image
    image_object_array[i] -> SetProperty, HIDE=1

    output_image = ( temporary(output_image) + rgb_image ) < 255

endfor

output_image = byte(temporary(output_image))
output_image = reform(output_image, 3, xpix, ypix, /overwrite)

return, output_image
```

=========================