Subject: Re: ROI stack
Posted by Chris[5] on Fri, 13 Jun 2008 00:01:33 GMT
View Forum Message <> Reply to Message

On Jun 12, 11:45 am, mega...@phas.ubc.ca wrote:
> Hello,
> Can anyone see where I'm going wrong here? I'm trying to generate a
> binary mask of a multislice MRI image. When I run this at the
> command line it works, but when I put it in a loop, it fails to return
> anything. i.e. all variables of interest are "undefined"
> Any help/tips would be appreciated!
> Thanks,
> Megan
> AT COMMAND LINE:
> xroi, img[*,*,i], regions_out = roi, /Block
> mask = roi -> ComputeMask(Dimensions = dims, Mask_Rule=2)
> masks[*,*,i] = mask
> LOOP:
> pro twodmask
> img = read_nifti(dialog_pickfile())
> dims = size(img[*,*,3], /dimensions)
> s = size(img, /dimensions)
> n_slices = 2 ;s[2]
> masks = fltarr(dims[0], dims[1], s[2])
> for i = 0,n_slices-1 do begin
> xroi, img[*,*,i], regions_out = roi, /Block
> mask = roi -> ComputeMask(Dimensions = dims, Mask_Rule=2)
> masks[*,*,i] = mask
> endfor
>
> end

When you run a procedure (something that starts with pro), all of the variables defined within the main body are local in scope. That is, when you get to the "end" statement, all of the variables within the program are discarded. Try this:

Change the first line from "PRO twodmask" to "function twodmask." Add the line "return, mask" on the line before "end." Call the function by typing in "mask=twodmask()." This now feeds the variable mask to you upon completion.

chris