Subject: Re: Dereferencing a Pointer Array
Posted by John-David T. Smith on Tue, 01 May 2001 19:28:14 GMT
View Forum Message <> Reply to Message

Art Burden wrote:

> I can't figure out a way to dereference the pointers to all 12 pixel
> values from a given coordinate in one step.

As you can imagine, it would be a sordid business indeed if you could dereference entire pointer arrays all at once:

parr=[ptr_new([1,2,3]),ptr_new([[5,6],[7,8]])]

print,total(*parr,1)

hmm.. what would we do with that? Since pointers can point to anything, pointer arrays don't necessarily point to similarly dimensioned blocks of data, or even similarly typed:

parr=[ptr_new([1,2,3]),ptr_new([[5,6],[7,8]]),ptr_new('Octop us')]

If you really know you will have N images of the same size, you can have your cake and eat it too. Instead of storing them as a pointer array, store them in a single pointer as an image cube:

pcube=ptr_new(randomu(sd,20,20))
*pcube=[[[*pcube],[[randomu(sd,20,20)]]]

now you can easily get the mean along the "depth" of the cube:

meanim=total(*pcube,3)/n_ims

This will be much faster than separately dereferencing/adding a long list of pointed-to arrays. It is less flexible though (what if you wanted to remove an array from the middle of the list?).

JD