Subject: Re: accessing large arrays quickly
Posted by D. Mattes on Thu, 27 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

well, this question about accessing large arrays has proven a rather embarrasing entry to this prestigious newsgroup, as i have discovered the real root of my problem. i was using the bytscl operation on each slice before drawing it to the window, thus slowing things down quite a bit. the performance hit was only noticeable once the array size grew larger than say 256x256x30, so i felt sure something else was culprit. but now, if i spend the time to bytscl the entire volume first, accessing the slices isn't putting any crimp in my style. thanks to those who took the time to respond!

what would be nice though, is a routine similar to tvscl, but with all the flexibility of bytscl, and with a couple extra features. tvscl only allows specifying a top value. it would be nice to specify a bottom value, to make working with split color tables a lot easier. does anybody know of such a procedure???

also, if anyone can answer this question, you get serious kudos! i am defining several objects and structures in my programs. if i add a tag to a structure once this structure has existed in memory, i must restart IDL and recompile, in order for my changes to be accepted. i always perform garbage collection on all the heap variables, using heap_gc. how can i get IDL to forget about all previous structure definitions???

thanks!
david mattes

On Thu, 27 May 1999, D. Mattes wrote:

> hello idl gurus: i have a very large volume array out of which i extract
> 3 orthogonal 2-d slices and display these slices in three separate
> windows. i extract a slice by assignment:
>
> ```
>         slice=data(*,*,zslice)
> ```
>
> then i scale the slice, and finally display it using tv. once the volume
> grows to larger than 10Meg, i suffer a performance hit on the array access
> times, and my image browser slows down considerably. how can i improve
> performance???
>
> some ideas i've had:
> 1. render the entire volume and specify cutting planes to just display
> the slice of interest.
> 2. use an external c function, like memcpy, to speed up the variable
> swapping when i assign 2-d array as a crosssection of the volume array.

> 3. store each possible slice separately, perhaps in a linked list.
>
> do you idl gurus out there have any suggestions or comments on my ideas???
>
> thank you in advance for your time.
>
> david mattes
>
>
>
>