Subject: Array Concatenation Optimization
Posted by KM on Tue, 08 Feb 2005 19:19:32 GMT
View Forum Message <> Reply to Message

I've read JD's tutorial
[http://www.dfanning.com/tips/array_concatenation.html] a few times,
but cannot get my array juggling as fast as I would like it...

I'm converting an 8 bit image to 24 bit RGB. Why? Well, long story,
but I am stuck in the Z buffer and need to make nice anti-aliased
color images. So everything is 3x as large&thick, converted to RGB,
and then rebined...

The conversion to RGB is currently one of the bottlenecks in my
code, and I would like to speed it up. I am using this function:

```
function toRGB, r,g,b, image
s = size(image,/dim)
image_rgb = bytarr( 3, s[0], s[1] )
image_rgb[0, *, *] = r[image]
image_rgb[1, *, *] = g[image]
image_rgb[2, *, *] = b[image]
return, image_rgb
end
```

I can cut the execution time in half if I change the entire function
to this one line:

```
return, [[[r[image]]],[[g[image]]],[[b[image]]]]
```

But now it is [n,m,3], and the WRITE_PNG procedure needs it to be
[3,n,m], and wrapping a TRANSPOSE() around that 1 line makes it go
from 2x as fast to 7x as slow as the original function.

I don't get any improvement if I change the * to explicit ranges,
although I remember reading that this should make array insertions
faster....

```
image_rgb[0, 0:s[0]-1, 0:s[1]-1] = r[image]
```

Another trick I have read about is to use the TEMPORARY() function,
but I don't think it is applicable in this case.

Any other suggestions?

Thanks,

Ken Mankoff

http://edgcm.org/
http://spacebit.dyndns.org/