Subject: Re: quick matrix algebra question
Posted by b_gom on Mon, 09 Jan 2006 21:29:29 GMT
View Forum Message <> Reply to Message

Thanks, I knew it would be something like this. Reading through the documentation for all the permutations of #, ##, matrix_multiply, and transpose was just too much for a monday morning.

Brad

Dick Jackson wrote:
> Hi Brad,
>
> <b_gom@hotmail.com> wrote in message
> news:1136835388.675373.55640@z14g2000cwz.googlegroups.com...
>> I'm sure someone must have a quick solution for this problem. I have a
>> color transformation that I want to apply to an RGB pixel as follows:
>>
>> M=fltarr(3,3)
>> RGB=bytarr(3)
>> ;fill RGB and M with some values
>> result = transpose(M ## RGB) ;apply the transform.
>
> To do this with consistency when there's more than one pixel, you might want
> to flip things around:
>
> IDL> m=FIndGen(3,3)
> IDL> RGB=BIndGen(3)
> IDL> Print,Transpose(M ## RGB)
> 5.00000 14.0000 23.0000
> IDL> Print,RGB ## Transpose(M)
> 5.00000 14.0000 23.0000
>
> Same result, but now this is extensible to a (3, n) array...
>
>> Now, result contains the new values in the same format as RGB.
>> The question is: how do I do this efficiently on an RGB image (ie a
>> (3,col,row) array)? Quick, before I use a for loop!
>
> col=40
> row=50
> RGB=BIndGen(3, col, row)
>
> ;; Reform RGB to (3, n)
>
> RGB = Reform(rgb, 3, Long(col)*row, /Overwrite) ; this is fast
>
> result = RGB ## Transpose(M) ;apply the transform.

>
> ;; Reform the arrays back to (3, col, row)
>
> RGB = Reform(RGB, 3, col, row, /Overwrite)
> result = Reform(result, 3, col, row, /Overwrite)
>
> There you go!
>
> Cheers,
> --
> -Dick
>
> Dick Jackson / dick@d-jackson.com
> D-Jackson Software Consulting / http://www.d-jackson.com
> Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392
