
Subject: Re: Accelerating a one-line program doing matrix multiplication

Posted by on Thu, 30 Sep 2010 16:38:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Sep 30, 5:03 pm, Paolo <pgri...@gmail.com> wrote:

> On Sep 30, 10:59 am, Paolo <pgri...@gmail.com> wrote:

>

>

>

>> [skip]

>

>>> FUNCTION vc2rc_accel, v0,v1,v2,v3,vc

>>> npoints = (SIZE(vc, /DIMENSIONS))[1]

>>> for i=0L, npoints-1 DO BEGIN

>>> vc[* ,i] = vc[0,i] * v1 + vc[1,i] * v2 + vc[2,i] * v3 + v0

>>> endfor

>>> RETURN, vc

>>> END

>

>> No, that's using a for loop - that's why it is slow.

>> You want something like this (no loops):

>

>> vc0=vc[0,*]

>> vc1=vc[1,*]

>> vc2=vc[2,*]

>> vc[0,*]=vc0*v1[0]+vc1*v2[0]+vc3*v3[0]

>> vc[1,*]=vc0*v1[1]+vc1*v2[1]+...

>> vc[2,*]=vc0*v1[2]+...

>

>> Ciao,

>> Paolo

>

> well, there's a bit of an index mismatch

> in there...

>

> vc[0,*]=vc0*v1[0]+vc1*v2[0]+vc2*v3[0]

> vc[1,*]=vc0*v1[1]+vc1*v2[1]+vc2*v3[1]

> vc[2,*]=vc0*v1[2]+vc1*v2[2]+vc2*v3[2]

Got it, thanks. I changed it in my test, it turned out to be slower than the original version (about 30% slower).

By now, seeing no way to improve the original version with an arbitrary vc, I stick to the version of "adding REBINS" for the case where all indexes from an image are given as input. This is the critical case for my application anyways. When a smaller region is wanted, the original version can be used.
