Subject: Re: Accelerating a one-line program doing matrix multiplication
Posted by Jeremy Bailin on Tue, 28 Sep 2010 15:31:07 GMT

View Forum Message <> Reply to Message

On Sep 27, 5:18 am, Axel M <axe...@gmail.com> wrote:

> Hi all,
>
> I wrote a one-line function to convert a list of points from "voxel
> coordinates" (image coordinates) to "real coordinates" (physical
> coordinates):
>
> ;input: the points "vc", the spatial origin of an image v0 and its x,
> y, and z orientation vectors (v1,v2,v3).
> FUNCTION vc2rc, v0,v1,v2,v3,vc
>   RETURN, [[v1],[v2],[v3]] # vc + REBIN(v0, SIZE(vc, /DIMENSIONS))
> END
>
> For example, I give the image coordinate [8,1,0] and I want as output
> something like [34.25, 4.12, 0], indicating the location of this voxel
> in space. And the same thing but, instead of having one input point,
> having several millions.
>
> The function looks simple to me and it works great. BUT, for large
> images (e.g. 500x500x200 voxels), it is terribly slow and uses way too
> much memory... Am I doing something wrong, could I save speed
> somewhere? I guess there should be some way to accelerate it, but I am
> not able to see how...
>
> I also have the opposite function, in my opinion also too slow (though
> faster than the other)...
>
> FUNCTION rc2vc_round, v0,v1,v2,v3,rc
>   RETURN, ROUND((rc - REBIN(v0, SIZE(rc, /DIMENSIONS))) ## INVERT([[v1],
> [v2],[v3]]))
> END
>
> I would be really grateful for any clue!

You can try adding /SAMPLE to the REBIN call, but I suspect that it's
not the bottleneck.

-Jeremy.