Subject: Re: efficient matrix multiplication
Posted by amaravad on Tue, 21 Mar 1995 21:49:53 GMT
View Forum Message <> Reply to Message

In article <D5LsIF.417@rockyd.rockefeller.edu>,
Darren Orbach <orbach@rockvax.rockefeller.edu> wrote:
> Greetings all.
>
> I need to find the inner product between a single floating point
> image (of size 64*48; let's call it specialimage) and several
> thousands of integer-type images (all with same 64*48 size).
> The integer images are stored in files, each of which consists
> of a stack of approximately 100 images (so the file structure
> is int(64, 48, 100). My task is to, as efficiently as possible,
> read in specialimage, read in the integer files, perform the inner
> product operation, and store the output into an array of floats.
>
> In order to avoid performing a multiplication for every individual
> snapshot, I've been replicating specialimage 100 times (so that it's
> now a floating point array of dim 64*48*100), reading
> in an entire image file into a variable (of size integer(64*48*100)),
> and then simply multipliying the two large arrays. This leaves me
> with a new array of size 64*48*100 (call it productarray).
>
> Here's the question: In order to extract the inner products from this
> big array, I've been looping over the z-variable and doing a "total()"
> operation: for i = 0, 99 do output(i)=total(productarray(*,*,i)).
> This process seems highly inefficient, and since I'm doing this
> hundreds of times every time I need the series of inner products,
> I'm looking for a better method. Is there a function analogous to
> total(), which can be directed to act over a specified dimension without
> looping over every element of that dimension? If it's at all relevant,
> I'm using PV-WAVE Advantage 5.0.
>
> Thanks in advance.

One thing you can do to reduce the number of float point operations,
is to quantize special image, ie factor specialimage into a floating
point constant times an integer array. Let SI = specialimage array.
FI = any of the integer arrays in file.

ie SI = q*NEWSI (q is the floating point quantization factor)
  SI*FI = q*NEWSI*FI
  total(SI*FI)=total(q*NEWSI*FI)=q*total(NEWSI*FI)
You are now doing only one fp op for each array, ofcourse
at the cost of some quantization error.

hope this helps. comments welcome.

--
Ratnakar Amaravadi Software/Hardware Engineer
Department of Radiology I.U. School of Medicine
(317)274-1843 (w) amaravad@indiana.edu
(317)274-4074 (fax) ratty@foyt.indyrad.iupui.edu