Subject: Re: computation time for convolution
Posted by wlandsman on Thu, 10 Jul 2014 12:16:56 GMT
View Forum Message <> Reply to Message

I knew that the OP was showing a special case but I was still finding CONVOL() to be faster than using an FFT. I now realize that the FFT compute time is independent of the relative size of the kernel and the image (since the kernel must be converted to the same size as the image prior to the FFT multiplication). So instead of varying the size of the image, I kept the image size fixed at 1024x1024 and varied the size of the kernel. The speed of CONVOL varies with the number of points in the kernel, while the FFT speed is almost indecent of the kernel size. For a small kernel (my usual case) CONVOL remains much faster. --Wayne

Kernel size: 16
convolve: 1.2780330
convol: 0.053172827
convol_fft: 1.2723532

Kernel size: 32
convolve: 1.2661600
convol: 0.20477700
convol_fft: 1.2720819

Kernel size: 64
convolve: 1.2892501
convol: 0.80016303
convol_fft: 1.2787650

Kernel size: 128
convolve: 1.2785149
convol: 2.8997500
convol_fft: 1.2978389

Kernel size: 256
convolve: 1.2797129
convol: 9.5446420
convol_fft: 1.2797601

Kernel size: 512
convolve: 1.3157580
convol: 22.437527
convol_fft: 1.3163621

Code:
```
pro test
a = randomn(seed,1024,1024)
for i=4,9 do begin
  b = dist(2^i)
  time0=systime(1)
```

```
  c1=convolve(a,b)
  time1=systime(1)
  c2=convol(a,b)
  time2=systime(1)
  c3=convolve(a,b)
  time3=systime(1)
  print,'Kernel size: ',2l^i
  print,'convolve:', time1-time0
  print,'convol:', time2-time1
  print,'convol_fft:', time3-time2

endfor
end
```

>
> You are not testing convol, you are testing a very special case (convol(a,a) calculates the sum in a single position, all other array elements are set to zero).
>
>
>
> You should use a more realistic kernel, eg b=dist(2l^(i-1)). With this I got:
>
>
>

```
>         1024x       1024
> 
> convolve:      2.4647841
> 
> convol:      35.345544
> 
> convol_fft:      2.8855970
```

>
>
>
> regards,
>
> Lajos