Subject: Re: fast convolving question
Posted by rogass on Fri, 30 May 2008 12:44:58 GMT
View Forum Message <> Reply to Message

Dear Chris,
thank you again for your reply and the amount of time you invested.

To understand, what I mean, it seems to be better to explain it for
very small matrices.

So, let's say you have a dist(3) kernel and a dist(7) matrix.
At first to overcome the problem with negative indices of the strict
numerical solution of convolving matrices, I padded the matrix in each
direction with 2 zeros, so the resulting matrix is now 9x9 (0,matrix,0
in x- and y-direction).

Then I pre-compute indices to speed up the process (main idea):

1.For the kernel: 0 - 8 + reform to vector

2.0. For the Matrix (first vector): 20-19-18-12-11-10-2-1-0(=indsmall)
+ reform to vector and insert it into matrix -
> mat(20-19-18-12-11-10-2-1-0 + ind(0)) <- (ind(0) is 0)

2.1. For the Matrix (second vector): 29-28-27-20-19-18-12-11-10 +
reform to vector and insert it into matrix -
> mat(20-19-18-12-11-10-2-1-0 + ind(1)) <- (ind(1) is 9)

till 2.48. 80-79-78....

3. As third step I multiply kernel-vector with the mat-vectors, so:

conv(0) = kernel ## mat(indsmall+ind(0))
conv(1) = kernel ## mat(indsmall+ind(1))
...
conv(48)= kernel ## mat(indsmall+ind(48))

4. Reform conv to 7x7 and return it

The trick is to only multiply the kernel as vector with the reformed
submatrix of the matrix. I tested all types of convolving - the above
code is only a snippet - and the fastest one were always my
unfortunately not right indexing no-for-loop.

Besides that strict convolving is a very simple scheme. Just
multiplying the always same kernel as vector with the
i.subarray(padded with zeros at the edges) of matrix(ixj) as vector

(beginning from down right to upper left) and repeating this ixj
times. Reform the given result back again to matrix.

But unfortunately, only the loop-method for k=0,48 do conv(k) = ...
works perfectly.

I found several methods to convolve discrete without any loops, but
they are always slower than fft or my one-loop-method, except the no-
loop-method which is more than 100 times faster than fft or convol.

So, please, please, please help me again and try to implement e.g.
indgen as the for-to-loop

Thanks and best regards

Christian