
Subject: Re: Avoiding FOR loops (version googleplex.infinity)

Posted by [Gaurav](#) on Thu, 10 Apr 2008 06:13:54 GMT

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

Dear All,

Over the past two days I was able to think of an algorithm and it solved my problem. And now, looking at Tom's solution above, it appears that it is almost the same thing that I came up with.

What I did was to pad up my image with $\text{kernelSize}/2$ pixels all around and then compare it with temporary arrays shifted all around the kernel by turn. Wherever the shifted array was equal to the padded array, I added one in the final, output array. Run this loop for the kernel size and you are done. Now it is up to you guys to decide if it is exactly the same as Tom's or there is some difference and as to which would be faster. Here follows my code:

```
##### Code begin #####
dims = size(img,/dimensions) ; getting the dimensions of input image
output_img = bytarr(dims(0), dims(1))

kernelSize = 5 ;define the kernel size (here, 5)
padSize = FLOOR(kernelSize /2.0) ;calculate the padding to generate
around the original image
paddedImg = bytarr(dims(0)+2*padSize, dims(1)+2*padSize) ;initialize
the padded image
paddedImg(padSize, padSize) = img ;define the padded image
templmg = bytarr(dims(0)+2*padSize, dims(1)+2*padSize) ;will contain
the padded, final output

for x = -padSize, padSize do begin
  for y = -padSize, padSize do begin
    if x eq 0 and y eq 0 then continue ;we do not want to compare with
the element itself
    ;the x, y loops shift the whole image array around the kernel and
compares the shifted image with the original
    indices = WHERE(paddedImg eq SHIFT(paddedImg, x, y)) ;
    templmg(indices) = finalmg(indices) + 1 ;increment '1' wherever an
equality is found
  endfor
endfor
output_img = EXTRAC(templmg, padSize, padSize, dims(0),
dims(1)) ;extract the output image from the padded output image.
##### end of code/algorithm #####
```

In my case it works wonderfully well and speeds things up by a factor of a few hundred times. Any further optimization would be highly appreciated.

I guess, I ought to go for a manicure for my gnawed nails now.

Cheers,
Gaurav
