
Subject: Re: speed up image processing
Posted by [Jye](#) on Fri, 24 Jul 2009 03:24:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

At first glance you could loop through the one dimensional subscripts.

```
FOR indx = 0, N_ELEMENTS(image)-1 DO image_out[indx] = min(image  
[vector_index+indx])
```

But as is the IDL way try to remove all loops. image_out1 in the below
pro takes 0.21s compared to 2.71 and 3.42 when a for loop is used.

PRO foo

```
image = DIST(100, 200)
```

```
s = SIZE(image, /dim)  
ncols = s[0]  
nrows = s[1]
```

```
vector_index = [1*ncols+1, 2*ncols+2, 3*ncols+3]
```

```
ind = INDGEN(s)  
t=systime(1)  
for k=0,99 do begin
```

```
    image_out1 = min([image[ind+vector_index[0]]], [image[ind  
+vector_index[1]]], [image[ind+vector_index[2]]], DIM=3)
```

```
endfor  
print, systime(1)-t
```

```
image_out2 = image  
t=systime(1)  
for k=0,99 do begin
```

```
    FOR indx = 0, N_ELEMENTS(image)-1 DO image_out2[indx] = min(image  
[vector_index+indx])
```

```
endfor  
print, systime(1)-t
```

```
image_out3 = image  
t=systime(1)  
for k=0,99 do begin
```

```
for j = 0, nrows-1 do begin
  for i = 0, ncols-1 do begin
    x = image(i, j)
    indx = j*ncols + i
    image_out3(i, j) = min(image[vector_index+indx])
  endfor
endfor
```

```
endfor
print, systime(1)-t
```

```
window, 0
tvsc1, image, 0
tvsc1, image_out1, 1
tvsc1, image_out2, 2
tvsc1, image_out3, 3
tvsc1, image_out1 - image_out3, 4
tvsc1, image_out2 - image_out3, 5
```

```
END
```
