



Efficient summed Area Table Calculation with Numpy

Ask Question

I'm trying to calculate a [summed area table](#) of a feature count matrix using python and numpy. Currently I'm using the following code:

```
def summed_area_table(img):  
    table = np.zeros_like(img).astype(int)  
  
    for row in range(img.shape[0]):  
        for col in range(img.shape[1]):  
            if (row > 0) and (col > 0):  
                table[row, col] = (img[row, col] +  
                                    table[row, col - 1] +  
                                    table[row - 1, col] -  
                                    table[row - 1, col - 1])  
            elif row > 0:  
                table[row, col] = img[row, col] + table[row - 1, col]  
            elif col > 0:  
                table[row, col] = img[row, col] + table[row, col - 1]  
            else:  
                table[row, col] = img[row, col]  
  
    return table
```

The above code takes about 35 seconds to perform the calculation on a 3200 x 1400 array. Is there any way to use Numpy trick to speed up the computation? I realize the fundamental speed problem lies in the nested python loops, but I don't know how to avoid them.

python

arrays

optimization

numpy

asked Aug 28 '14 at 21:03



Nick

18 ● 3

1 Answer

¿No encuentras la respuesta? [Pregunta en](#)



There's a NumPy function `cumsum` for cumulative sums. Applying it twice yields the desired table:

```
import numpy as np

A = np.random.randint(0, 10, (3, 4))

print A
print A.cumsum(axis=0).cumsum(axis=1)
```

Output:

```
[[7 4 7 2]
 [6 9 9 5]
 [6 6 7 6]]
[[ 7 11 18 20]
 [13 26 42 49]
 [19 38 61 74]]
```

Performance analysis:

(<https://stackoverflow.com/a/25351344/3419103>)

```
import numpy as np
import time

A = np.random.randint(0, 10, (3200, 14))

t = time.time()
S = A.cumsum(axis=0).cumsum(axis=1)
print np.round_(time.time() - t, 3),
```

Output:

```
0.15 sec elapsed
```

edited May 23 '17 at 12:31



Community ♦

1 • 1

answered Aug 28 '14 at 21:09



Falko

10.7k • 12 • 35 • 76