

TABLA DE RESULTADOS **ARQUITECTURA DE SISTEMAS DISTRIBUIDOS. 3º GTI.** **PRÁCTICA 2: ACELERACIÓN DE FILTROS DE IMÁGENES.** **ALUMNO: Alejandro Fernández Trigo**

OBJETIVOS Y PREPARACIÓN.

- Esquema del código C de un desenrollado de alto nivel del bucle elegido.

- bucles
+ instr.

```
for(int x=1; x<(img_width-1); x++){
    int sum0, sum1, sum2 = 0;
    sum0 += ptr[3*x + 0]; // +0, +1, +2
    ...
    sum0 += ptr[3*(x-1) + 0]; // +0, +1, +2
    ...
    sum0 += ptr[3*(x+1) + 0]; // +0, +1, +2
    ...
    sum0 += ptr[3*x + 3*img_width + 0]; // +0, +1, +2
    ...
```

```
sum0 += ptr[3*(x-1) + 3*img_width + 0]; // +0, +1, +2
...
sum0 += ptr[3*(x+1) + 3*img_width + 0]; // +0, +1, +2
...
ptr[3*x + 0] = sum0/n
ptr[3*x + 1] = sum1/n
ptr[3*x + 2] = sum2/n
}
// Fin
```

¿Se podría desenrollar teóricamente todo el procesamiento del fotograma?

➔ No; hay instrucciones ajenas al bucle, que no pueden desenrollarse.

2.1 Código desenrollado

	CvSmooth	asd_blurring_unrolled
Tiempo con /O2	0,00278134	0,00186372
Aceleración entre CvSmooth y unrolled con /O2	1.0	$A = \frac{0,002...}{0,001...} = 1,492 \approx 49\% \uparrow$

Contestar: ¿A qué bucles sustituyen estas líneas de código?

```
sum += ptr[3*x+color];
sum += ptr[3*(x-1)+color];
...
sum += ptr[3*(x+1)-3*image_width+color];
```

```
➔ for(int yshift = -kernel_radius; yshift <= kernel_radius; yshift++) {
    for(int xshift = -kernel_radius; xshift <= kernel_radius; xshift++) {
        sum += ptr[...];
    }
}
```

INTENTANDO MEJORAR LAS PRESTACIONES:

Optimización /Od

	CvSmooth	asd_blurring_unrolled	NUEVO CÓDIGO ALUMNO asd_blurring_unrolled_own
Tiempo mínimo por fotograma	0,002783	0,0129645	0,0102552
Aceleración entre CvSmooth y nuevo asd_blurring_unrolled_own	$A = 0,01... / 0,0027... \approx 3,68 \approx 200\% \uparrow$		
Aceleración entre unrolled y nuevo asd_blurring_unrolled_own	$A = 0,01... / 0,0102... \approx 1,26 \approx 26\% \uparrow$		

CvSmooth es más rápido

Optimización /O2

	CvSmooth	asd_blurring_unrolled	NUEVO CÓDIGO ALUMNO asd_blurring_unrolled_own
Tiempo mínimo por fotograma	0,00278134	0,001836372	0,000678636
frames por segundo (fps) alcanzados (como máximo)	359	536	1473
Aceleración entre CvSmooth y nuevo asd_blurring_unrolled_own	$A = 0,002... / 0,0006... \approx 4,1 \approx 300\% \uparrow$		
Aceleración entre unrolled y nuevo asd_blurring_unrolled_own	$A = 0,001... / 0,0006... \approx 2,75 \approx 100\% \uparrow$		

Justifique las diferencias que observa en los resultados entre la versión unrolled que se da y la versión que Ud. ha desarrollado asd_blurring_unrolled_own para la opción de compilación /O2: Piense quien cambia más: CPI o Ninst

● Desarrollando el código se obtienen mejores resultados ya que aunque aumenta el no de instrucciones, se evitan los saltos, y esto reduce el CPI. La versión "own" está más desarrollada, de ahí sus mejores resultados en el tiempo.

Justifique las diferencias que observa en los resultados entre la versión unrolled que se da y la versión que Ud. ha desarrollado asd_blurring_unrolled_own para la opción de compilación /O2

● Ante un mayor desarrollo, el compilador ha encontrado las mejores instrucciones que hacen la ejecución más óptima posible frente a la versión unrolled provista (menos desarrollada).

1/2 →