
Sistemas Operativos

Departamento de Ingeniería en Informática

LAB1: programación con hebras Pthreads

1 Objetivo

El objetivo de este laboratorio es conocer, usar y apreciar las características de programación multi-hebrea usando Pthreads. En este laboratorio usted implementará un programa multi-hebrea de procesamiento de imágenes.

2 Operadores morfológicos

En el ámbito del procesamiento de imágenes, la morfología consiste de un conjunto de operadores matemáticos que se aplican sobre imágenes. En morfología estos operadores se conocen por operadores morfológicos y generalmente se especifican como un patrón bidimensional binario o **elemento de estructuración (ES)**. Por ejemplo, la figura 1 muestra dos ES comunes de 3×3 píxeles:

0	1	0
1	1	1
0	1	0

1	1	1
1	1	1
1	1	1

Figure 1: Dos elementos de estructuración.

Todo ES tiene un centro. En las figuras anteriores, el centro es indicado con el pixel en gris. Como el ES se aplica a otra imagen, el centro del ES, indica sobre qué pixel de la imagen se está aplicando el ES, y por lo tanto el correspondiente pixel de la imagen resultante. Como se puede ver, un ES define una vecindad (o conjunto de píxeles) de la imagen original a los cuales se les aplicará alguna operación; estos píxeles corresponden a los píxeles donde el ES es igual a 1 (blanco).

Los operadores morfológicos comunes son:

1. Erosion: adelgaza el foreground
2. Dilation: expande el foreground
3. Closing: elimina hoyos en el foreground
4. Opening: elimina pixeles ruido del foreground

En este trabajo sólo implementaremos el operador **Dilation** (\oplus). Este operador se define matemáticamente como sigue. Sea A una imagen binaria y B un ES, entonces

$$C = A \oplus B \quad (1)$$

$$C = \bigcup_{p \in A} B_p \quad (2)$$

Una forma equivalente de definición es:

$$C = A \oplus B \quad (3)$$

$$C = \bigcup_{p \in B} A_p \quad (4)$$

Esta última definición es más apropiada para entender cómo implementar dilation. El resultado se puede interpretar como la unión (suma lógica) de los valores de los píxeles de la imagen original (A) que *están debajo del ES* (B) para todos los píxeles del ES. Por ejemplo, considere el ES que define la cruz, aplicada a una imagen binaria, como lo muestra la figura 2. En la imagen se ha representado con dos tonalidades de gris el ES y la imagen, pero en ambos casos el valor es 1. Los píxeles no mostrados tienen valor 0, es decir negro (para mostrar las imágenes en papel se ha optado por mostrarlas con los colores invertidos). En el primer ejemplo, la unión da como resultado un valor vacío; en cambio en el segundo ejemplo, da como resultado un conjunto no vacío. Recuerde que la unión es sólo de los píxeles de la imagen de entrada; los píxeles del ES no participan de la unión y sólo definen qué píxeles de la imagen de entrada se consideran en la operación. Luego, el resultado en el primer ejemplo es un pixel con valor 0, y en el segundo un valor 1. El resultado siempre se almacena en el pixel de la imagen de salida, indicado por el centro del ES.

En un programa computacional, y cuando la imagen está representada con 0s y 1s, la operación de unión puede implementarse como la búsqueda del valor máximo. Entonces, en el primer ejemplo el máximo entre valores 0s es 0; en el segundo caso el máximo entre 0s y 1s es 1.

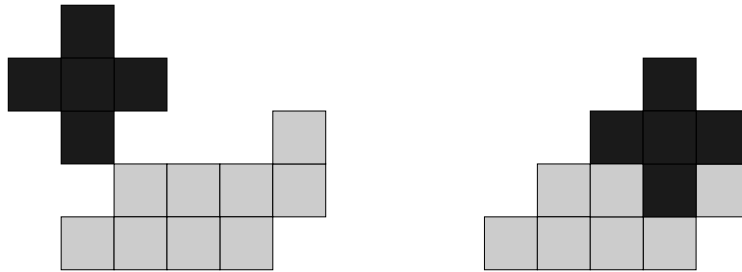


Figure 2: Aplicación del ES a dos posiciones distintas de la imagen.

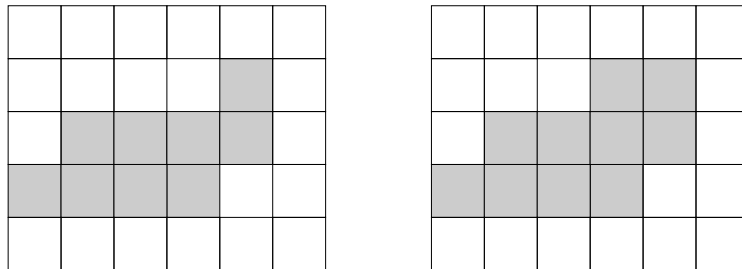


Figure 3: Resultado de dilation en dos posiciones distintas de la imagen.

La operación de dilation debe ser aplicada a cada pixel de la imagen de entrada.

3 Estrategia multi-hebras

Para este lab implementaremos dilation sólo con el ES que define la cruz para imágenes binarias. La estrategia de división del trabajo consistirá en usar un número variable de hebras, tal que cada una de ellas aplique el operador a un subconjunto de filas. Es decir, la hebra h_i operará desde la fila s_i (start) hasta la e_i (end). Es bastante sencillo calcular s_i y e_i para una imagen de $N \times N$ píxeles, cuando trabajan H hebras. Usted debe asegurar que todas las hebras realicen (aproximadamente) el mismo trabajo. Cuando H divide exactamente a N , todas las hebras realizarán la misma cantidad de trabajo, pero sino, una o varias realizarán más trabajo que otras. Por ejemplo, si $N = 15$ y $H = 3$, todas las hebras operarán sobre 5 filas de la imagen. Pero si $N = 16$ y $H = 3$, una de ellas lo hará sobre una fila más que el resto.

Note que para realizar la división del trabajo, es necesario que las hebras tengan un ID correlativo desde 0 hasta $H - 1$. El ID entregado por `pthread_create()` no nos sirve en este caso.

4 El programa y la imagen

Diseñe y escriba un programa multi-hebreado en C que implemente la operación de dilatación. El programa deber ser invocado de la siguiente forma:

```
$ ./dilation -i imagen_entrada.raw -O imagen_salida.raw -N ancho_imagen  
-H numero_hebras -D opcion_debug
```

Es requerimiento de este lab el uso de `getopt()` para manejar argumentos de línea de comando. Documentese!

La imagen estará almacenada en disco en formato binario, es decir una secuencia de números enteros almacenados según representación binaria. Un pixel puede tomar sólo dos valores, 0 (negro), o 1 (blanco). Como no se conoce el largo de la imagen, usted debe usar procuramiento dinámico y usar los llamados al sistema `open()`, `read()`, `write()`, y `close()` para manejar el I/O del programa, hacia y desde las imágenes. La opción `-D` indica si se imprime (1) o no (0) por stdout la imagen resultante en términos de ceros y unos. Esta opción es útil para verificar, con imágenes pequeñas (por ejemplo 10×10), que el programa está funcionando.

Anteriormente se dijo que el operador de dilation debe ser aplicado a cada posición de la imagen de entrada. Para no tener problemas con los píxeles de los bordes, sólo aplicaremos la operación a los píxeles entre $[1, 1]$ y $[N - 2, N - 2]$.

5 Entregables

Tarree, comprima y envíe a `luis.loyola@usach.cl` al menos los siguientes archivos:

1. `Makefile`: archivo para make que compila ambos programas
2. `dilation.c`: archivo con el código. Puede incluir otros archivos fuentes.

Fecha de entrega:
domingo XXX de mayo antes de las 24:00 hrs.

6 Ejemplo

Un simple ejemplo de dilatación se muestra a continuación, para el ES de la cruz. Nuevamente, recuerde que solo por motivos de impresión en papel, se muestran la imágenes negadas.

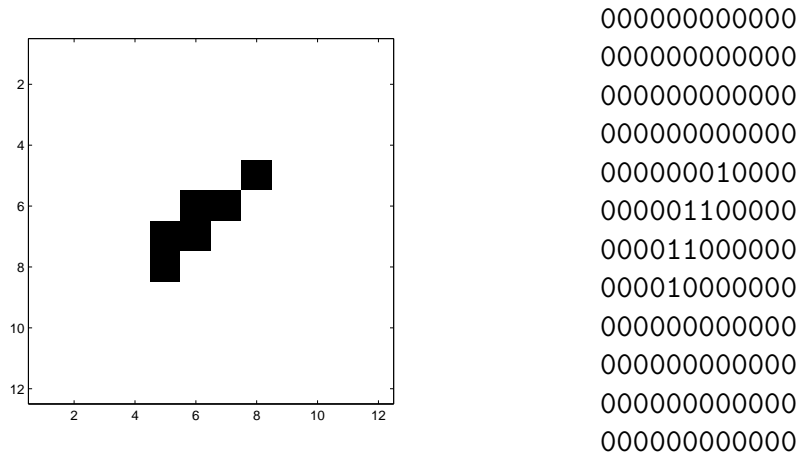


Figure 4: Imagen de entrada.

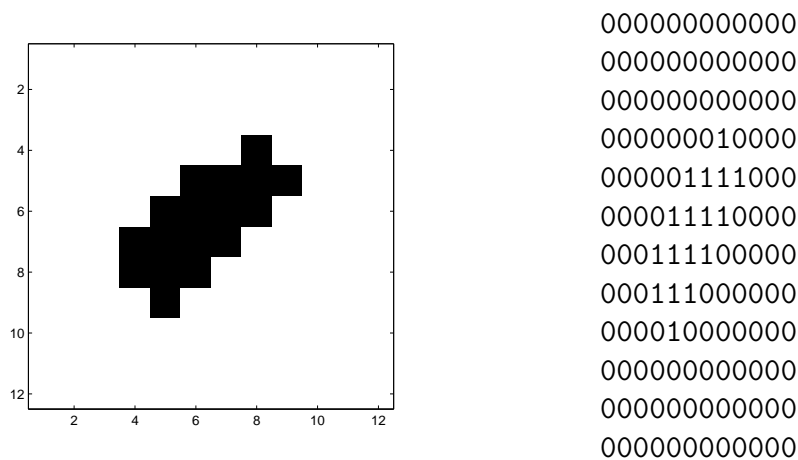


Figure 5: Imagen de salida.