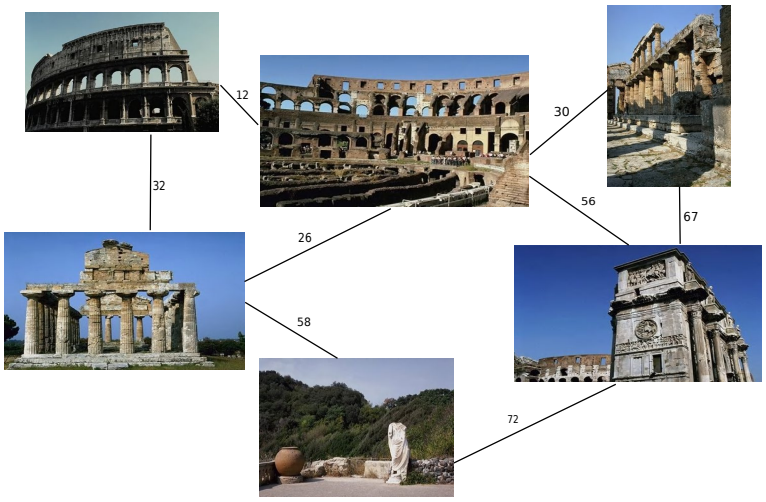
GRAFO DE IMAGENES

Concurso de Estructuras de Datos y Algoritmos

curso 2010-2011

1. Introducción

El objetivo del concurso de este año es el de calcular las distancias (disimilitudes) que existen entre un conjunto de imágenes provisto. Cada imagen pasará a estar representada como un nodo en un grafo por lo que la distancia entre ambas imágnes se calculará como la distancia (el camino de mínimo coste) que hay entre dichos nodos en el grafo.

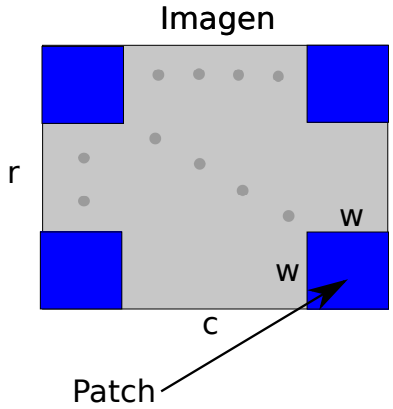


Tanto la representación de las imágenes, la definición de la distancias entre ellas así como las restricciones a tener en cuenta en la construcción del grafo son aspectos que se tratan en las siguientes secciones.

2. Representación de imágenes. Definición de Patch

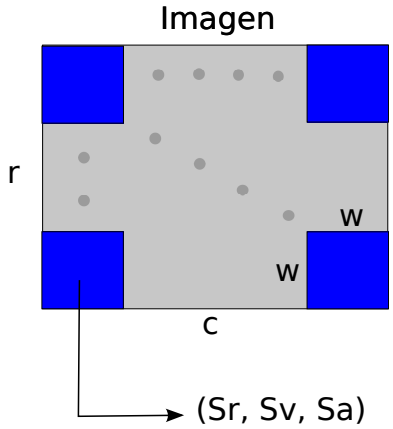
A las imágenes a representar se les va a hacer una modificación previa a la representación. Dicha modificación consiste en que los valores de rojo, verde y azul de cada pixel deberán ser divididos (división entera sin redondeo) por un parámetro c (cuantificación). Por ejemplo si $c = 32$ el valor de rojo $r = 153$ pasará a ser $r = 4$. Este proceso de cuantificación se realizará una única vez cuando se lea la imagen. Dicho parámetro c será un valor entero, potencia de 2 en el intervalo $[1, 128]$.

Una vez realizada esta cuantificación la representación de las imágenes estará basada en un conteo de características locales o *patches*. Un patch no es más que una parte de la imagen definida por un cuadrado de $w \times w$ píxeles. De una imagen extraeremos todos los posibles patches moviendo dicha ventana de $w \times w$ píxeles a lo largo y ancho de toda la imagen:

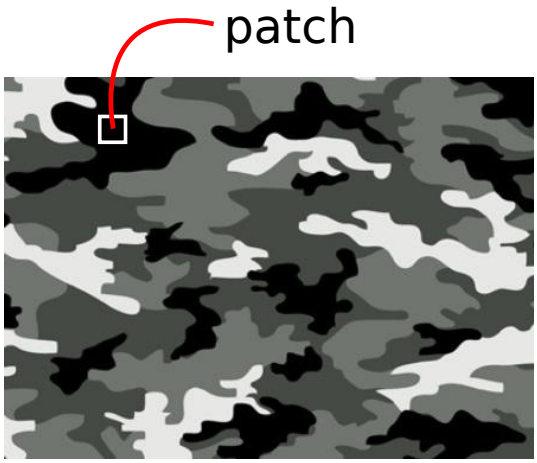


En general de una imagen de r filas y c columnas ($r \times c$ píxeles) podremos extraer $((r-w)+1) \times ((c-w)+1)$ patches de tamaño $w \times w$.

A su vez un patch queda representado por únicamente 3 valores, la suma de la componente roja (Sr), verde(Sv) y azul (Sa) de todos los píxeles que lo componen.



Claramente en una misma imagen podremos tener muchos patches cuya representación con estos tres valores coincida. Por ejemplo en la siguiente imagen es fácil comprobar que el patch seleccionado tendrá valores idénticos a los de otros muchos patches que se pueden extraer de la misma imagen:



Finalemente la imagen pasará a estar representada por una lista de todos los patches que lo componen y para cada patch el número de veces que aparece en la imagen (cuenta):

$$imagen = \{(Sr, Sv, Sa), cuenta, (Sr, Sv, Sa), cuenta, \ldots, (Sr, Sv, Sa), cuenta\}$$

Lógicamente imágenes diferentes pueden tener diferente número de patches.

3. Distancia entre imágenes

Una vez definida la representación de las imágenes pasamos a definir cómo calcular la distancia entre dos imágenes así definidas. Para tal fin se hace necesario introducir algo de notación que facilite la definición de dicha distancia. Sea:

- P_I el conjunto de patches de la imagen I
- N_{Iq} la cuenta asociada al patch q en la imagen I. Cuántas veces aparece el mismo patch q en la imagen I
- C_{IJ} el conjunto de patches comunes de las imágenes I y J, lógicamente $C_{IJ} = C_{JI}$
- D_{IJ} el conjunto de patches que aparecen en la imagen I pero no en la J, $D_{IJ} = P_I - C_{IJ}$
- D_{JI} el conjunto de patches que aparecen en la imagen J pero no en la I, $D_{JI} = P_J - C_{IJ}$
- $abs(\cdot)$ es la función valor absoluto.

Presentada ya la notación a emplear, se define la distancia entre dos imágenes I y J como:

$$d(I,J) = \sum_{q \in D_{IJ}} N_{Iq} + \sum_{q \in D_{JI}} N_{Jq} + \sum_{q \in C_{IJ}} abs(N_{Iq} - N_{Jq}) \tag{1}$$

Por lo tanto, la distancia entre dos imágenes I y J se define como la suma de las cuentas de los patches que están en I pero no en J *más* la suma de las cuentas de los patches que están en J pero no en I *más* la suma para todos los patches comunes del valor absoluto de la diferencia de las cuentas. Esta definición de distancia cumple la propiedad simétrica, $d(I,J) = d(J,I)$

Veamos un ejemplo de este cálculo de distancia,

- Sea la imagen I:
  $\{(100,200,100), 34, (200,200,100), 27, (32,32,32), 25\}$
- Sea la imagen J:
  $\{(100,200,100), 15, (0,0,0), 31, (12,100,25), 45, (200,200,100), 40\}$.

Tendríamos entonces:

- $C_{JI} = \{(100,200,100), (200,200,100)\}$
- $D_{IJ} = \{(32,32,32)\}$
- $D_{JI} = \{(0,0,0), (12,100,25)\}$

Por lo que,

$$d(I,J) = (25) + (31+45) + (abs(34-15) + abs(27-40)) = 133 \tag{2}$$

4. Construcción del grafo

Para la construcción del grafo se tendrán en cuenta las distancias entre las imágenes. Cada imagen quedará representada como un nodo del grafo. Cada imagen se conectará con las k imágenes de menor distancia (en caso de empate de distancia se escogeran las imágenes por orden de lectura del fichero de entrada). Dicha conexión será una arista bidireccional cuyo peso es la distancia entre ambas imágenes.

Hay que tener en cuenta que es mas que posible que de una imagen salgan (o incidan) más de k aristas. Estas aristas así definidas serán las únicas presentes en el grafo, el resto de aristas se supondrán de peso infinito indicando que no hay camino directo entre dos nodos (imágenes) del grafo. Además, dependiendo del valor de k y las imágenes, es muy probable que no exista ningún camino en el grafo que una dos imágenes en particular.

Con el grafo así definido, redefinimos la distancia entre dos imágenes como el coste del camino de menor coste que las une en el grafo. Si dicho camino no existe dicha distancia será infinito (inf).

5. Entrada y Salida

La entrada al programa a realizar será un fichero con la lista de imágenes a procesar, el parámetro k, el parámetro w y el parámetro c:

```
$ concurso lista.txt 1 3 64
```

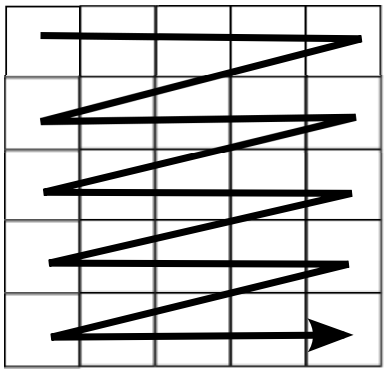
El fichero "lista.txt"tiene por cada linea un nombre (y extensión) de una imagen:

```
imagen1.ppm
imagen2.ppm
imagen3.ppm
imagen4.ppm
imagen5.ppm
```

En este ejemplo se asume que las imágenes están en el mismo directorio donde se está ejecutando el programa.
El formato de las imágenes en PPM ascii es el siguiente:

```
P3
384 256
255
45 88 156  47 92 159  50 95 162  47 96 162  46 95 161  46 97 162
48 98 161  49 99 162  54 100 162  55 101 163  60 101 163  61 102 164
62 104 164  62 104 164  60 105 162  60 105 162  60 105 162  61 106 163
62 107 166  62 107 166  62 107 166  62 107 166  61 107 167  62 108 168
61 110 169  61 110 169  62 112 171  63 113 172  65 115 174  65 115 174
65 117 175  65 117 175  67 116 174  65 114 172  65 114 172  66 115 173
...
```

La primera linea contiene un código (P3) que indica que el formato es PPM ascii. La segunda linea contiene el número de columnas (384) y filas (256) de la imágen. La tercera línea contiene el valor máximo con el cual se va a representar los colores (de 0 a 255). Por último aparece una serie de tripletes (45 88 156) que corresponden al nivel de rojo, verde y azul de cada uno de los píxeles. Recordar que estos valores tendrán que ser divididos por el parámetro c (división entera sin redondeo). En el fichero los píxeles aparecen en el siguiente orden, fila a fila de la imagen (de arriba a abajo) y dentro de cada fila de izquierda a derecha tal y como muestra la figura:



La salida del programa será una matriz en ascii con los valores de todas las distancias **en el grafo** entre todas las imágenes:

```
d(1,1) d(1,2) .... d(1,n)
d(2,1) d(2,2) .... d(2,n)
....
d(n,1) d(n,2) .... d(n,n)
```

Por supuesto las distancia $d(I, I)$ para toda imagen I es 0. Así mismo $d(I, J) = d(J, I)$ dado que todas las aristas son bidireccionales. En el caso que no exista un camino en el grafo entre las imágenes I,J el valor a escribir por la salida es inf. Un ejemplo de salida podría ser este:

```
0 12 inf
12 0 37
inf 37 0
```