

Cálculo del histograma de una imagen

Una imagen digital se puede representar por una matriz de tamaño $N \times M$ elementos dónde N es el número de filas y M es el número de columnas. En cada uno de estos elementos se almacena el nivel de gris correspondiente a esa posición.

El histograma asociado a una imagen contiene el número de píxeles que tienen el mismo nivel de gris. Puede entenderse como la probabilidad de que un valor de gris determinado aparezca en la imagen.

Una vez visto lo que es una imagen digital y su histograma, se pide un programa que permita calcular y dibujar el histograma asociado a una imagen. Esta imagen esta almacenado en un fichero con el siguiente formato:

```
N_filas N_columnas N_niveles_gris
Datos.....
```

A continuación se muestra un ejemplo de fichero que contiene una imagen:

```
20 20 10
1 2 3 4 5 6 7 8 9 0 0 9 8 7 2 3 2 3 1 5
1 2 4 5 6 4 5 4 3 2 0 9 8 7 2 3 2 3 1 5
0 9 8 7 2 3 2 3 1 5 1 2 4 5 6 4 5 4 3 2
0 9 8 7 2 3 2 3 1 5 1 2 4 5 6 4 5 4 3 2
0 9 8 7 2 3 2 3 1 5 1 2 4 5 6 4 5 4 3 2
0 9 8 7 2 3 2 3 1 5 1 2 4 5 6 4 5 4 3 2
0 9 8 7 2 3 2 3 1 5 0 9 8 7 2 3 2 3 1 5
1 2 3 4 5 6 7 8 9 0 0 9 8 7 2 3 2 3 1 5
1 2 3 4 5 6 7 8 9 0 0 9 8 7 2 3 2 3 1 5
1 2 3 4 5 6 7 8 9 0 0 9 8 7 2 3 2 3 1 5
1 2 3 4 5 6 7 8 9 0 0 9 8 7 2 3 2 3 1 5
1 2 3 4 5 6 7 8 9 0 0 9 8 7 2 3 2 3 1 5
1 2 3 4 5 6 7 8 9 0 0 9 8 7 2 3 2 3 1 5
1 2 3 4 5 6 7 8 9 0 0 9 8 7 2 3 2 3 1 5
1 2 3 4 5 6 7 8 9 0 0 9 8 7 2 3 2 3 1 5
1 2 3 4 5 6 7 8 9 0 0 9 8 7 2 3 2 3 1 5
1 2 3 4 5 6 7 8 9 0 0 9 8 7 2 3 2 3 1 5
1 2 3 4 5 6 7 8 9 0 0 9 8 7 2 3 2 3 1 5
```

El tamaño de la imagen así como el número de niveles de gris aparece en la primera línea de fichero.

Para almacenar la imagen y el histograma se deben utilizar vectores dinámicos puesto que esa información se encuentra almacenada en el fichero y no se conoce a priori.

```
#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>
#include <string>

//Definicion de constantes
const int ESCALA = 5;
//Definicion de tipos
typedef int* fila;
typedef fila* matriz;
typedef int* histograma;

//Prototipos de funciones

//Leer de la entrada estandar
void leer_imagen(matriz mat, int& filas, int& column, int&
n_niveles);

//Leer de fichero
void f_leer_imagen(ifstream& f_in, matriz mat,int filas,
int column);

//Imprimir por pantalla
void imprimir_imagen(const matriz mat, int filas, int
column);

//imprimir a fichero
void f_imprimir_imagen(ofstream& f_out, const matriz, int
filas, int column);

//Calculo el histograma
void calcular_histograma(const matriz mat, int filas, int
column, histograma hist, int niveles);

//Gráfica que representa el histograma
void imprimir_histograma(const histograma hist, int
niveles);
```

```
//Programa principal

int main()
{
    matriz imagen;
    histograma hist;
    int fil,col, niveles;
    int i;
    string nombre_in;
    ifstream ent_f;
    ofstream sal_f;

    cout << "Introduce mi nombre de fichero que contiene
la imagen\n";
    //Leo nombre y almaceno en una variable tipo 'string'
    cin >> nombre_in;

    //Abro el fichero
    ent_f.open( nombre_in.c_str() );

    //Compruebo si se pudo abrir el fichero
    if (!ent_f)
        cout << "Error abriendo fichero de datos."<< endl;
    else
    {
        //Procedo a leer el fichero
        //leo tamaño a reservar y niveles
        ent_f >> fil >> col >> niveles;

        //Reservo espacio para imagen
        imagen = new fila [fil];
        for ( i = 0 ; i < fil ; i++)
            imagen[i] = new int [col];

        //Leo los datos del fichero
        f_leer_imagen(ent_f,imagen,fil,col);

        //Cierro el fichero
        ent_f.close();

        //Imprimo la imagen leida
        imprimir_imagen(imagen,fil,col);

        //Reservo espacio para el histograma dinamicamente
        hist = new int [niveles];
    }
}
```

```
//Calculo del histograma

calcular_histograma(imagen,fil,col,hist,niveles);

//Muestro el histograma calculado
imprimir_histograma(hist,niveles);

//Destruyo la imagen
for ( i = 0 ; i < fil ; i++)
    delete imagen[i] ;
delete imagen;
//Destruyo el histograma
delete histograma;

}
system("PAUSE");
return 0;
}
```

```
//Implementación de funciones
void leer_imagen(matriz mat, int& filas, int& column, int&
n_niveles)
{
    int i,j;

    cout << "Introduce el tamaño de la matriz (filas
columnas):\n";
    cin >> filas;
    cin >> column;
    cin >> n_niveles;

    for(i = 0; i < filas; i++)
        for (j = 0; j < column ; j++)
        {
            cout << "Elemento :" << i << ' ' << j << endl;
            cin >> mat[i][j];
        }

    return ;
}
//leer de fichero la imagen
void f_leer_imagen(ifstream& f_in, matriz mat, int filas,
int column)
{
    int i,j;
    //Leo de fichero

    for(i = 0; i < filas; i++)
        for (j = 0; j < column ; j++)
            f_in >> mat[i][j];
    return;
}
//Imprimo por pantalla
void imprimir_imagen(matriz mat, int filas, int column)
{
    int i , j;

    for(i = 0; i < filas; i++)
    {
        for (j = 0; j < column ; j++)
            cout << mat[i][j] << ' ';
        cout << endl;
    }

    return ;
}
}
```

```
//Calculo el histograma
void calcular_histograma(const matriz mat, int filas, int
column, vector hist, int niveles)
{
    int i ,j;
    int valor;

    //Inicializo los contadores
    for ( i = 0; i< niveles; i++)
        hist[i]=0;
    //Recorro la matriz para contar valores repetidos
    for (i = 0; i < filas ; i++)
        for( j = 0; j < column ; j++)
        {
            valor = mat[i][j];
            if ( valor < niveles)
                hist[valor]++;
        }

    return;
}
//dibujo el histograma por pantalla
void imprimir_histograma(const vector hist, int niveles)
{
    int i, j;
    int asteriscos;
    //Imprimo los valores del histograma escalados segun
    ESCALA
    cout << "DIBUJO DE HISTOGRAMA:" << endl;
    for (i = 0; i < niveles ; i++)
    {
        cout << hist[i] << ": ";
        asteriscos = int(hist[i] / ESCALA);
        for ( j = 0; j < asteriscos ; j++)
            cout << '*';
        cout << endl;
    }
    return;
}
```