

Ecualización

En la siguiente actividad se estudiará la ecualización. Como pudo observar en la experiencia anterior, el histograma de una imagen está conformado por la frecuencia con la que se encuentran las intensidades de color en una imagen, cuyos valores de intensidad pueden ir desde 0 hasta 255. De esta forma se puede obtener un histograma en RGB conformado por 3 matrices cuya interacción genera una imagen a color. Es posible traspasarla a escala de grises generando sólo una matriz, a través de una transformación numérica de los valores RGB, tal como se muestra en la ecuación (1).

$$EG = (R*30 + G*59 + B*11) / 100 \quad (1)$$

Ecualización

Es una transformación de histograma en donde se busca repartir de forma uniforme las intensidades de éste en todo el rango de grises. Es utilizada una matriz escalonada $f [0...255]$ obtenida de la ecuación (2).

$$F(a) = \text{histograma_acumulado} * 255 / M \times N \quad (2)$$

Donde $M \times N$ son las dimensiones de la matriz Imagen.

Desarrollo

Lea la guía con atención antes de escribir el código.

1. Cree un nuevo proyecto de Windows Forms en lenguaje C# y agregue dos botones en la ventana de diseño:
 - a. Cargar imagen.
 - b. Limpiar.
2. Agregue las siguientes bibliotecas:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;
```

3. Configure el botón "Cargar Imagen":
 - i. Cree una variable del tipo *OpenFileDialog* y llame al método *ShowDialog()*.

- ii. Cree un *Bitmap* con la dirección obtenida del método *FileName* de la variable declarada en (i.).

```
private void button1_Click(object sender, EventArgs e)
{
    OpenFileDialog abrir = new OpenFileDialog();
    abrir.ShowDialog();
    Bitmap Imagen = new Bitmap(abrir.FileName);           //Creo Bitmap
}
```

4. Comience el recorrido de la matriz extrayendo los valores de intensidad por pixel y conviértalos a escala de grises:
 - i. Utilice el método *GetPixel(x,y)* (con x e y, coordenadas del pixel a ser leído) que le devuelve un valor de tipo *Color*.
 - ii. Aplique la ecuación (1) para la obtención de la matriz en escala de grises, en donde los valores de las matrices RGB son obtenidos por *Pixel.R*, *Pixel.G* y *Pixel.B* respectivamente. Declare dicha matriz y llénela con los valores obtenidos.
 - iii. Cree la matriz *histograma[256]* y llénela con la frecuencia con que se repiten los valores de intensidades en la imagen.

```
-----Definiciones-----
int[,] matriz = new int[Imagen.Width, Imagen.Height]; // Defino matriz para guardar valores de pixeles obtenidos
int[] histograma = new int[256];                      //Defino histograma

-----Comienza recorrido de la matriz de la imagen-----

for (int f = 0; f < Imagen.Width; f++)                //Empiezo a recorrer matriz de imagen
{
    for (int c = 0; c < Imagen.Height; c++)
    {
        Color pixel = Imagen.GetPixel(f, c);          //Extraigo información por pixel
        int Grisescala = (pixel.R * 30 + pixel.G * 59 + pixel.B * 11) / 100; //Convierto a escala de grises
        matriz[f, c] = Grisescala;
        histograma[Grisescala] = histograma[Grisescala] + 1;
    }
}
```

5. Configure el ecualizado de la imagen.
 - a. Cree una variable de tipo *int* en donde vaya guardando el valor acumulado del histograma.
 - b. Cree una matriz en donde guardará el valor de la ecuación (2).
 - c. Desarrolle la ecuación (2) para todo el recorrido del histograma y guarde el valor obtenido en la matriz creada.

```
int[] ecualizacion = new int[256];                    //Defino nuevo histograma para ecualizado
int suma = new int { };
```

```
-----Ecualizado-----  
  
suma = histograma[0];  
ecualizacion[0] = 0;  
for (int a = 1; a < 255; a++)  
{  
    ecualizacion[a] = suma * 255 / (Imagen.Width * Imagen.Height);  
    suma = suma + histograma[a];  
}  
ecualizacion[255] = 255;
```

6. Luego cree la matriz de la imagen ecualizada. Declare la matriz y recorra por todo el ancho y alto de la imagen original, guardando el nuevo valor por pixel. Además de esto genere el histograma de la nueva imagen.

```
int[,] matrizec = new int[Imagen.Width, Imagen.Height];//// Defino matriz para guardar valores ecualizados  
int[] hecua = new int[256];
```

```
-----Creo imagen ecualizada-----  
  
for(int f=0; f<Imagen.Width;f++)  
{  
    for(int c=0;c<Imagen.Height;c++)  
    {  
        matrizec[f, c] = ecualizacion[matriz[f, c]];  
        hecua[ecualizacion[matriz[f, c]]] = hecua[ecualizacion[matriz[f, c]]] + 1;  
    }  
}
```

7. Agregue 3 *PictureBox* a la ventana de diseño y 2 *chart*. En la ventana de diseño, modifique la propiedad del *PictureBox* "SizeMode" a Zoom.
8. Muestre las imágenes en los *PictureBox*.
- Defina 2 *bitmap* para guardar la imagen en escala de grises y la ecualizada, respectivamente. Ambas con las mismas dimensiones de la imagen original.
 - Recorra la imagen y defina el color en escala de grises como un RGB con las tres matrices iguales.
 - Guarde el color de cada pixel, para cada imagen.
 - Finalmente las imprima en los *PictureBox*.

```
Color gris = new Color();  
Color ecua = new Color();
```

```
-----Muestro imagenes en pictureBox-----  
  
Bitmap EscalaGris = new Bitmap(Imagen.Width, Imagen.Height); //Creo nuevo bitmap para guardar imagen en escala de grises  
Bitmap Ecualizada = new Bitmap(Imagen.Width, Imagen.Height); //Creo nuevo bitmap para guardar imagen ecualizada  
  
for (int f=0;f<Imagen.Width;f++)  
{  
    for(int c=0; c<Imagen.Height;c++)  
    {  
        gris = Color.FromArgb(matriz[f, c], matriz[f, c], matriz[f, c]);  
        ecua = Color.FromArgb(matrizec[f, c], matrizec[f, c], matrizec[f, c]);  
        EscalaGris.SetPixel(f, c, gris);  
        Ecualizada.SetPixel(f, c, ecua);  
    }  
}  
pictureBox1.Image = Imagen;  
pictureBox2.Image = EscalaGris;  
pictureBox3.Image = Ecualizada;
```

9. En los cuadros chart, imprima los histogramas creados:

```
//-----Imprimo histogramas-----  
  
chart1.ChartAreas[0].AxisX.Minimum = 0;  
chart1.ChartAreas[0].AxisX.Maximum = 255;  
  
chart2.ChartAreas[0].AxisX.Minimum = 0;  
chart2.ChartAreas[0].AxisX.Maximum = 255;  
  
for (int a = 0; a < 256; a++)  
{  
    string[] hist = new string[256];  
    hist[a] = a.ToString();  
    Series intensidades = chart1.Series.Add(hist[a]);  
    intensidades.Points.AddXY(a, histograma[a]);  
  
    Series intensidades2 = chart2.Series.Add(hist[a]);  
    intensidades2.Points.AddXY(a, hecua[a]);  
}
```

10. En el botón Limpiar, configure el vaciado de las gráficas *Chart*, de modo que sea posible cargar otra imagen.

```
1 referencia  
private void button2_Click(object sender, EventArgs e)  
{  
    chart1.Series.Clear();  
    chart2.Series.Clear();  
}
```