

Tarea 17

Angel Manrique Pozos Flores; N.C M07211505
Tecnológico Nacional de México,
Blvd. Industrial, Mesa de Otay, 22430 Tijuana, B.C., México.

5 de mayo de 2016

Implementar la matriz de co-ocurrencia en 0 grados con distancia 1 y calcular al menos 2 medidas de textura.

1. Matriz de co-ocurrencia

La matriz de co-ocurrencia describe la frecuencia de un nivel de gris que aparece en una relacion espacial especifica con otro valor de gris, dentro del area de una ventana determinada, dicha matriz puede considerarse como un resumen de la forma en que los valores de los pixeles ocurren al lado de otro valor en una pequeña ventana.

Normalmente el procedimiento para la generacion de imagenes de textura requiere que se definan cinco variables:

- Tamaño de la ventana.
- Banda espectral de la ventana.
- Las derivadas de las texturas.
- Numero de bits del canal de salida.
- La componente espacial, que corresponde a la distancia interpixel y el angulo para el computo de la co-ocurrencia.

El tamaño de la ventana debe ser cuadrada y con un numero impar de pixeles. El resultado del calculo de la textura es un valor unico que representa a la ventana completa.

Dicho valor es colocado en el lugar del pixel central, luego la ventana se mueve un pixel y el calculo se repite calculando una nueva matriz de co-ocurrencia para esta nueva ventana, obteniendo un nuevo valor que se coloca en el pixel central de la nueva posicion, de este modo se construye toda una nueva imagen con valores de texturas, como se muestra en la figura 1.

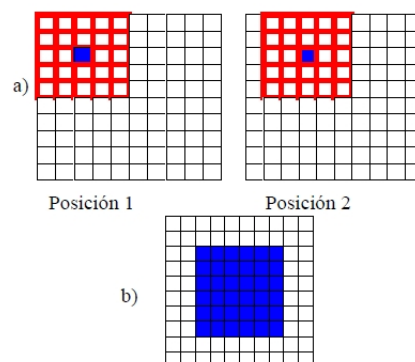


Figura 1: a) Se muestra sobre una imagen de dimensiones 10x10 píxeles, una ventana móvil de 5x5 en dos posiciones y el pixel central que recibe el resultado. b) Los píxeles que reciben el resultado y las 2 filas y columnas del borde de la imagen que reciben el valor de textura calculados de los píxeles cercanos que corresponden a la ventana.

Cada celda de la ventana debe situarse en una celda que este ocupada en la imagen original, esto significa que el pixel central de la ventana no puede ocupar un borde de la imagen, si una ventana tiene $N \times N$ dimensiones, una franja de $(N-1)/2$ píxeles alrededor de la imagen permanecerá sin resultados, usualmente los píxeles de borde representan una pequeña fracción de las imágenes por lo cual es un problema menor. Sin embargo, si la imagen es muy pequeña o la ventana muy grande, este efecto debe considerarse en el análisis de los resultados. Una forma de solucionar este problema es llenar esas celdas con el valor calculado para el pixel mas cercano.

Por otra parte el tamaño relativo de la ventana y de los objetos en la imagen determinan la utilidad de esta medida para la clasificación, es recomendable que la ventana sea menor que el objeto y lo suficientemente grande como para capturar la variabilidad del mismo.

Por ejemplo, en un bosque la textura esta determinada por las luces y las sombras de las copas, una ventana con el tamaño de un solo árbol no medirá la textura

del bosque, otra ventana cubriendo todo el bosque y los campos vecinos a el, tampoco mediran la textura del mismo.

La relacion espacial entre el pixel de referencia y su vecino puede ser en cualquiera de las 8 direcciones (Norte, Sur, Este y Oeste y las cuatro diagonales) pero solo se toman cuatro ya que existen opuestas y en vez de contarlas por separado existen formas mas sencillas de medirlas (matriz simetrica).

Cuando hablamos de la relacion *espacialmente invariable* se eligen las cuatro direcciones. N, NE, E, SE y se promedian, en forma angulas se puede decir como 0° , 45° , 90° y 135° como se muestra en la figura 2.

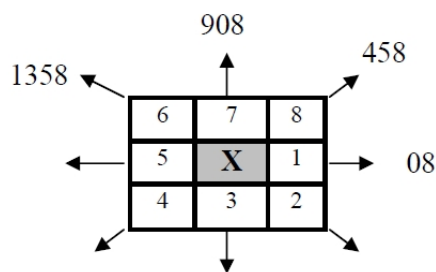


Figura 2: Los 8 vecinos alrededor del pixel X de acuerdo al angulo utilizado en el calculo de la matriz de co-ocurrencia.

Todas estas variables podrian generar muchos datos: asumiendo que se usan 7 medidas de textura o mas, 6 bandas espectrales, tamaños de ventanas que pueden variar entre 3x3 y 21x21, 3 niveles de cuantizacion del canal de salida 8, 16 y 32 bits y cuatro direcciones posibles, el resultado serian mas de 5000 imagenes de textura para una sola aplicacion, por ello no existe una forma de predecir con certeza cuales medidas seran las de mayor utilidad, sin embargo, existen algunos conceptos de importancia que hay que considerar.

- La inspeccion visual de algunas bandas de entrada pueden ayudar a determinar no usarlos, osea, usar bandas especificas como el rojo o el infrarrojo, el analisis de las componentes principales tambien puede ayudar a reducir las bandas de entrada.
- La inspeccion visual de una imagen tambien puede mostrar alguna direccionalidad, si no la hubiese, la mejor opcion es usar la espacialmente invariante.
- Muchas de las medidas texturales estan correlacionadas entre si.
- El mejor tamaño de la ventana se podra estimar inspeccionando visualmente la imagen.

El problema radica al seleccionar un conjunto de variables de textura, en general, la textura es una variable específica para cada imagen, donde el uso exitoso del análisis textural en una aplicación no implica necesariamente que se pueda aplicar globalmente, por ello, la selección de las variables de textura debería basarse en el estudio iterativo del conjunto imagen particular y condiciones del objeto a estudiar.

1.1. Cálculo de la matriz de co-ocurrencia

Como se observa en la figura 3, se hace la representación de una imagen prueba con sus correspondientes niveles a escala de grises, donde la imagen tiene 4 píxeles de lado y 4 niveles de grises, que son 0, 1, 2, 3.

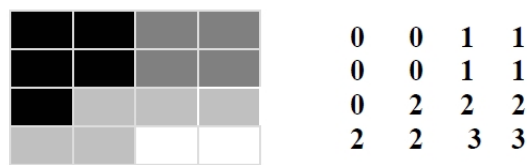


Figura 3: Se muestra una imagen de prueba de dimensiones 4x4 píxeles y 4 valores de grises 0, 1, 2, 3.

Esta matriz de co-ocurrencias considera la relación espacial entre dos píxeles, conocidos como *pixel de referencia* y *pixel vecino*, por ejemplo, si escogemos el pixel vecino que está situado un pixel a la derecha de cada pixel de referencia, esto lo podemos expresar como (1,0) : 1 pixel en la dirección x y 0 pixel en la dirección y.

Cada pixel en dicha ventana tomará la posición sucesivamente del pixel de referencia, empezando por el ubicado arriba a la izquierda y finalizando abajo a la derecha. Los píxeles ubicados en el margen derecho de la imagen original, no tienen un vecino ubicado a la derecha por ello no son utilizados.

Podemos utilizar entonces, diferentes relaciones entre píxeles, por ejemplo:

- (-1, 0) Un pixel a la izquierda del pixel de referencia.
- (1, 1) Un pixel a la derecha y un pixel abajo.

Las mayormente utilizadas son las relaciones que corresponden a las cuatro orientaciones principales.

- $(1, 0)$ → Orientacion horizontal correspondiente a 0°
- $(0, -1)$ → Orientacion vertical correspondiente a 90°
- $(1, 1)$ → Orientacion diagonal hacia arriba correspondiente a 45°
- $(1, -1)$ → Orientacion diagonal hacia abajo correspondiente a 135°

A continuacion se presentan las posibles combinaciones de niveles de grises para la imagen de prueba.

Pixel Vecino Pixel de Referencia	0	1	2	3
0	(0,0)	(1,0)	(2,0)	(3,0)
1	(1,0)	(1,1)	(2,1)	(3,1)
2	(2,0)	(1,2)	(2,2)	(3,2)
3	(3,0)	(1,3)	(2,3)	(3,3)

Figura 4: Aqui se muestran todas las posibles combinaciones de los 4 niveles de gris de la imagen de prueba.

La primera celda debe ser llenada con la cantidad de veces que ocurre dicha combinacion, en nuestro caso $(0, 0)$. Existen por lo tanto diferentes matrices de co-ocurrencia para cada relacion espacial, segun se considere el vecino de arriba, a un lado o en diagonal ($0^\circ, 45^\circ, 90^\circ$ o 135°) en la figura 5 mostramos la matriz de co-ocurrencia para la imagen de prueba para la orientacion $(1, 0)$ o orientacion horizontal.

2	2	1	0
0	2	0	0
0	0	3	1
0	0	0	1

Figura 5: Aqui se muestran todas las posibles combinaciones de los 4 niveles de gris de la imagen de prueba.

La primera celda debe completarse con la cantidad de veces que ocurre la combinacion $(0, 0)$ cuantas veces, en el area de la ventana un pixel con valor de

gris igual a 0 (un pixel vecino), esta situado a la derecha de otro pixel con valor 0 (pixel de referencia), en el caso de tener una orientacion horizontal de 0° , que corresponde con la relacion espacial (1,0).

Existen entonces distintas matrices de co-ocurrencia para cada region espacial, segun se considere el vecino ya sea en la posicion de arriba, abajo o diagonal. En la figura 6 podemos observar esta matriz de co-ocurrencia.

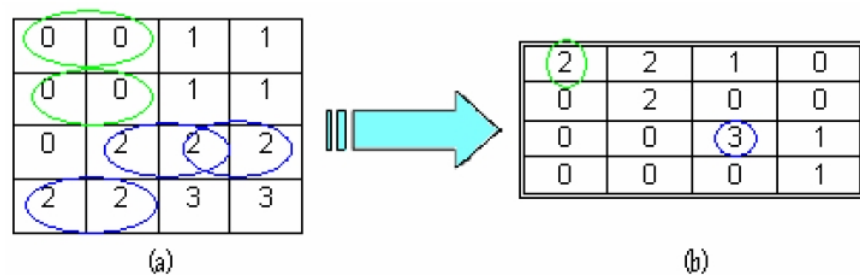


Figura 6: En a) se muestra la matriz de valores para y encerrados en verde y azul se muestra la relacion espacial para (1, 0).

b) Muestra la matriz de co-ocurrencias correspondiente a la matriz de coeficientes.

Esta matriz se interpreta de la siguiente forma, en la imagen de prueba, dos veces el pixel de referencia es 0, y su vecino a la derecha tambien es 0 y tres veces el pixel de referencia es 2 y su vecino de la derecha es 2, en la matriz b) se va haciendo un conteo de las veces que se repite.

Esta simetria es necesaria para el calculo, la cual se logra si cada par o distancia definida se cuentan 2 veces, una vez hacia su derecha y otra hacia la izquierda, una forma mas sencilla de lograr esto es sumar la transpuesta de esta a la misma matriz para asi obtener la tabla que se muestra en la figura 6.

4	2	0	0
2	4	0	0
0	0	6	1
0	0	1	2

Figura 7: Se muestra la matriz simetrica para una relacion espacial horizontal.

Una vez que hemos obtenido esta matriz simetrica, el paso siguiente es expresar esta matriz como probabilidad, donde la definicion mas simple de probabilidad viene dada por la ecuacion 1.

$$C_{i,j} = \frac{V_{i,j}}{\sum_{i,j=0}^{N-1} V_{i,j}} \quad (1)$$

Que se puede interpretar como, “*El numero de veces que un evento ocurre, dividido entre la sumatoria del total de eventos*”.

Donde:

- i es el numero de filas y j el numero de columnas.
- V es el valor de la celda (i, j) en la ventana.
- $C_{i,j}$ es la probabilidad en la celda (i, j) .
- N es el numero de filas o columnas.

Considerando la imagen de prueba de 4x4 pixeles y la relacion espacial $(1, 0)$, el numero total de posibles pares es 12, como se muestra en la figura 8 y para una relacion horizontal de derecha- izquierda ese numero se duplica.

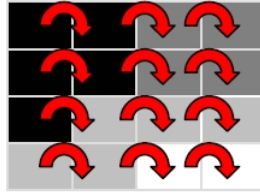


Figura 8: Total de pares posibles para una relacion espacial 1,0 si es una matriz de 4x4.

La ecuacion 1 transforma la matriz de co-ocurrencia en una tabla de aproximaciones, a este proceso se le conoce como *normalizacion de la matriz*, donde aplicamos dicha ecuacion a la matriz mostrada en la figura 7, donde la sumatoria de todos los elementos debe ser igual a 1 ya que esta normalizada.

1.2. Características de la matriz

La matriz de co-ocurrencia normalizada presenta ciertas características:

- Los elementos de la diagonal representan pares de pixeles que no tienen diferencias en su nivel de gris, si estos elementos tienen probabilidades grandes, entonces la imagen no muestra mucho contraste, la mayoría de los pixeles son identicos a los vecinos.

0.166 (4/24)	0.083 (2/24)	0.042 (1/24)	0 (0/24)
0.083	0.166	0	0
0.042	0	0.250	0.042
0	0	0.042	0.083

Figura 9: Matriz de probabilidades normalizada.

- Sumando los valores de la diagonal tenemos la probabilidad que un pixel tenga el mismo nivel de gris que su vecino.
- Las líneas paralelas a la diagonal, separadas una celda, representan los pares de pixeles con una diferencia de 1 nivel de gris, de la misma manera, sumando los elementos separados por dos celdas de la diagonal, tendríamos los pares de pixeles que presentan dos valores de gris de diferencia, a medida que nos alejamos de la diagonal la diferencia entre niveles de grises es mucho mayor.
- Sumando los valores de estas diagonales paralelas, obtenemos la probabilidad de que un pixel tenga 1, 2, 3... n niveles de grises de diferencia con su vecino.

Y las principales propiedades de la matriz de co-ocurrencia son:

- **Cuadrada:** El rango de los niveles de gris de los pixeles de referencia y el de los vecinos es el mismo por lo tanto las filas y las columnas tienen identico numero.
- **Presentan el mismo numero de filas y columnas que el numero de bits de la imagen:** La imagen de prueba tiene solo 4 valores posibles (0, 1, 2, 3) que corresponden a una matriz de 4x4. Los datos de 8 bits tienen 256 posibles niveles de gris, entonces la matriz de co-ocurrencias es de 256x256.
- **Matriz es simetrica con respecto a la diagonal:** Como se menciono anteriormente una matriz es simetrica cuando los mismos valores ocurren en las celdas opuestas a la diagonal, es decir, el valor en la celda 3,2 debe ser el mismo que el valor en la celda 2,3 para que la matriz sea considerada simetrica.

1.3. Medidas de textura

Una vez construida esta matriz y que ha sido normalizada pueden derivarse distintas medidas, a continuacion se explican algunos de estos descriptores texturales:

1. **Inercia o contraste:** Es una medida de la variacion local en una imagen, alcanza un valor alto cuando la imagen tiene mucho contraste y tiene un valor bajo, cuando los valores altos de la matriz estan cerca de la diagonal principal.

$$C_1 = \sum_{i,j} (i - j)^2 \cdot C_{i,j} \quad (2)$$

2. **Energía o momento angular de segundo orden:** Esta medida da valores altos, cuando la matriz de co-ocurrencia tiene pocas entradas de gran magnitud y es baja cuando todas las entradas son similares.

$$C_2 = \sum_{i,j} C_{i,j}^2 \quad (3)$$

3. **Entropia:** Este descriptor de textura, nos mide la aleatoriedad de la imagen, alcanzando su maximo valor cuando todos los elementos de la matriz de co-ocurrencia son iguales.

$$C_3 = \sum_{i,j} -C_{i,j} \cdot \log_2[C_{i,j}] \quad (4)$$

Se asume que $0 * \log(0) = 0$.

4. **Homogeneidad:** Es lo opuesto al contraste y se calcula mediante la ecuacion:

$$C_4 = \sum_{i,j} \frac{C_{i,j}}{1 + (i - j)^2} \quad (5)$$

La homogeneidad es alta cuando la matriz de co-ocurrencia se concentra a lo largo de la diagonal, esto va a ocurrir cuando la imagen es localmente homogenea.

5. **Probabilidad Maxima:** Se obtiene con

$$C_5 = \text{Max}\{C_{i,j}\} \quad (6)$$

6. **Tonalidad:** Se calcula la media de la matriz de co-ocurrencia, tanto para filas como para columnas, el valor del pixel no es ponderado por su frecuencia si no por la frecuencia de su co-ocurrencia en combinacion de un determinado valor del pixel vecino.

$$\begin{aligned} M_x &= \sum_{i,j} i \cdot C_{i,j} \\ M_y &= \sum_{i,j} j \cdot C_{i,j} \end{aligned} \quad (7)$$

Entonces la tonalidad la podemos definir como:

$$C_6 = \sum_{i,j} (i - M_x + j - M_y)^3 \cdot C_{i,j} \quad (8)$$

7. **Importancia:** Medida similar a la tonalidad que se define como:

$$C_7 = \sum_{i,j} (i - M_x + j - M_y)^4 \cdot C_{i,j} \quad (9)$$

8. **Medida de informacion de la correlacion:** Haralick propone la expresion:

$$C_8 = \frac{C_3 - H_{x,y}}{\text{Max}(H_x, H_y)} \quad (10)$$

Donde:

$$\begin{aligned} S_x(i) &= \sum_j C_{i,j} \longrightarrow H_x = - \sum_i S_x(i) \cdot \log_2[S_x(i)] \\ S_y(i) &= \sum_i C_{i,j} \longrightarrow H_y = - \sum_j S_y(i) \cdot \log_2[S_y(i)] \\ H_{x,y} &= - \sum_{i,j} C_{i,j} \cdot \log_2[S_x(i) \cdot S_y(j)] \end{aligned} \quad (11)$$

9. **GLCM media:** Se calcula con la siguiente ecuacion:

$$C_9 = \sum_{i,j=0}^{N-1} i P_{i,j} \quad (12)$$

10. **Dismilaridad:** Similar al contraste, es alta cuando la region tiene un contraste alto.

$$C_{10} = \sum_{i,j=0}^{N-1} C_{i,j} |i - j| \quad (13)$$

Estos pesos crecen linealmente a medida que nos alejamos de la diagonal (0, 1, 2, 3...n) el valor absoluto de $(0 - 0) = (1 - 1) = 0$ que son los valores de la diagonal.

Este nos mide la semejanza de la imagen consigo misma, algunas de sus propiedades son:

- Un objeto tiene mas alta correlacion dentro de si mismo que entre los objetos adyacentes.
- Pixeles cercanos estan mas correlacionados entre si , que los pixeles mas distantes.

El vector de características de co-ocurrencia estara formado por 96 elementos, a partir de cada sub imagen de detalles se calculan sus cuatro GLCM una para cada una de las orientaciones espaciales (0° , 45° , 90° y 135°) de dichas matrices GLCM correspondientes de la sub-imagen se pueden obtener las medidas de textura.

2. Desarrollo

Se utilizo la imagen de la figura 10 con su correspondiente matriz de niveles de gris (0, 1, 2, 3), para la realización de las pruebas.

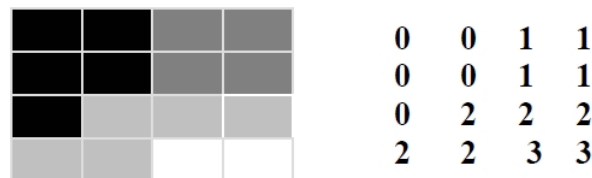


Figura 10: Matriz de niveles de gris utilizada en las pruebas para generar la matriz de co-ocurrencias.

Para ello se desarrollo el siguiente código en C++ utilizando las librerías de visión de OpenCV.

```

#include <opencv2/opencv.hpp>
#include <opencv2/core/core.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <stdio.h>
#include <iostream>
#include <vector>
#include <cmath>

using namespace cv;
using namespace std;

Mat img, gray, prob, homo, homo_p, cont, cont_p, disim
    , disim_p, glcmm, ener, entro, entro_p;
double maxval, minval;
double total = 0;
double temp = 0.00;
double den, homogeneidad, contraste, disimilaridad,
    glcmmedia, energia, entropia;
char* source_window = "Imagen Original";

template<typename T>
T limit_precision(T val, int precision) {
    return std::floor((val * std::pow(10,
        precision) + 0.5)) / std::pow(10, precision
    );
}

int main()
{
    /*
    img = (Mat_<uchar>(7, 7) <<
    3,1,3,4,5,6,8,
    4,1,1,2,4,5,7,
    4,2,1,1,3,3,6,
    4,3,1,1,1,2,4,
    4,4,2,1,1,1,1,
    7,6,5,3,1,1,1,
    8,7,6,5,2,3,5);*/

    img = (Mat_<uchar>(4, 4) <<

```

```

0, 0, 1, 1,
0, 0, 1, 1,
0, 2, 2, 2,
2, 2, 3, 3);

```

```

cout << "Imagen" << endl << img << endl << endl;

```

```

minMaxLoc(img, &minval, &maxval);
Mat g0(maxval + 1, maxval + 1, CV_8UC1, Scalar(0));
Mat g90(maxval + 1, maxval + 1, CV_8UC1, Scalar(0));

```

```

/// Construcccion GLCM de filas y columnas

```

```

for (int j = 0; j < g0.rows; j++)
{
    for (int i = 0; i < g0.cols; i++)
    {
        for (int l = 0; l < img.rows; l++)
        {
            for (int k = 0; k < img.cols; k++)
            {

                /// 0 Grados izquierda , derecha con d=1
                if ((k + 1) < img.cols)
                {
                    if (((img.at<uchar>(l, k) == j) && (img.at<uchar>
                        >(l, k + 1) == i)))
                    {
                        g0.at<uchar>(j, i) += 1;
                        total++;
                    }

                    if ((img.at<uchar>(l, k) == i) && (img.at<uchar>
                        >(l, k + 1) == j))
                    {
                        g0.at<uchar>(j, i) += 1;
                    }
                }
            }
        }
    }
}

```

```

/// 90 Grados arriba , abajo con d=1
if ((l + 1) < img.rows)

```

```

    {
        if (((img.at<uchar>(l, k) == j) && (img.at<
            uchar>(l + 1, k) == i)))
        {
            g90.at<uchar>(j, i) += 1;
            total++;
        }

        if ((img.at<uchar>(l, k) == i) && (img.at<uchar>
            >(l + 1, k) == j))
        {
            g90.at<uchar>(j, i) += 1;
        }
    }
}
}
}

```

```

/// Print GLCM
cout << "GLCM" << endl << "Grados = 90" << g90 << endl
    << endl;
cout << "GLCM" << endl << "Grados = 0" << g0 << endl
    << endl;

```

```

/// C_i, j
prob = g0.clone();
prob.convertTo(prob, CV_32FC1);

```

```

for (int j = 0; j < prob.rows; j++)
{
    for (int i = 0; i < prob.cols; i++)
    {
        temp = prob.at<float>(j, i) / total;
        prob.at<float>(j, i) = temp;
        prob.at<float>(j, i) = limit_precision((prob.at<
            float>(j, i)), 3);
    }
}

```

```

/// Print Ci,j
cout << "Matriz de Probabilidades" << endl << prob <<
    endl << endl;

/// Medidas de Textura
homo = prob.clone();
homo_p = prob.clone();
cont = prob.clone();
cont_p = prob.clone();
disim = prob.clone();
disim_p = prob.clone();
glcmm = prob.clone();
entro = prob.clone();
entro_p = prob.clone();
ener = prob.clone();
pow(ener,2,ener);

for (int j = 0; j < prob.rows; j++)
{
    for (int i = 0; i < prob.cols; i++)
    {
        ///Homogeneidad
        den = (1 + ((i - j)*(i - j)));
        homo_p.at<float>(j, i) = ((1 + ((i - j)*(i - j)))
            / 10.0);
        homo.at<float>(j, i) = (homo.at<float>(j, i) / den
            );
        homogeneidad += homo.at<float>(j, i);

        ///Contraste
        cont_p.at<float>(j, i) = (i - j)*(i - j);
        cont.at<float>(j, i) = (cont.at<float>(j, i)*
            cont_p.at<float>(j, i));
        contraste += cont.at<float>(j, i);

        ///Disimilaridad
        disim_p.at<float>(j, i) = abs(i - j);
        disim.at<float>(j, i) = (disim.at<float>(j, i)*
            disim_p.at<float>(j, i));
        disimilaridad += disim.at<float>(j, i);
    }
}

```

```

///GLCM media
    glcmm.at<float>(j , i) = (glcmm.at<float>(j , i)*i);
    glcmmedia += glcmm.at<float>(j , i);

///Energia
    energia += ener.at<float>(j , i);

///Entropia
    if (entro.at<float>(j , i)>0) //log of zero goes to
        infinity
    {
        entro_p.at<float>(j , i) = log(entro.at<float>(j ,
            i));
    }
    entro.at<float>(j , i) = entro.at<float>(j , i)*
        entro_p.at<float>(j , i);
    entropia += entro.at<float>(j , i);
}
entropia = -1* entropia;

///Forma matricial
cout << "Matriz de Pesos de Homogeneidad" << endl <<
    homo_p << endl << endl;
cout << "Matriz de Homogeneidad" << endl << homo <<
    endl << endl;
cout << "Matriz de Pesos de Contraste" << endl <<
    cont_p << endl << endl;
cout << "Matriz de Contraste" << endl << cont << endl
    << endl;

///Calculo
cout << "Energia: " << energia << endl << endl;
cout << "Contraste: " << contraste << endl << endl;
cout << "Homogeneidad: " << homogeneidad << endl <<
    endl;
cout << "Disimilaridad: " << disimilaridad << endl <<
    endl;
cout << "Entropia: " << entropia << endl << endl;
cout << "GLCM Media: " << glcmmedia << endl << endl;

```



```

imshow(source_window , img);
waitKey(0);
return 0;
}

```

obteniendo:

```

Imagen
[0, 0, 1, 1;
 0, 0, 1, 1;
 0, 2, 2, 2;
 2, 2, 3, 3]

GLCM
Grados = 90
[5, 0, 2, 0;
 0, 4, 2, 0;
 2, 2, 2, 2;
 0, 0, 2, 0]

Grados = 0
[4, 2, 1, 0;
 2, 4, 0, 0;
 1, 0, 6, 1;
 0, 0, 1, 2]

C_i,j o Matriz de Probabilidad
[0.167, 0.082999997, 0.041999999, 0;
 0.082999997, 0.167, 0, 0;
 0.041999999, 0, 0.25, 0.041999999;
 0, 0, 0.041999999, 0.082999997]

Matriz de Pesos de Homogeneidad
[0.1, 0.2, 0.5, 1;
 0.2, 0.1, 0.2, 0.5;
 0.5, 0.2, 0.1, 0.2;
 1, 0.5, 0.2, 0.1]

Matriz de Homogeneidad
[0.167, 0.041499998, 0.0083999997, 0;
 0.041499998, 0.167, 0, 0;
 0.0083999997, 0, 0.25, 0.021;
 0, 0, 0.021, 0.082999997]

Matriz de Pesos de Contraste
[0, 1, 4, 9;
 1, 0, 1, 4;
 4, 1, 0, 1;
 9, 4, 1, 0]

Matriz de Contraste
[0, 0.082999997, 0.168, 0;
 0.082999997, 0, 0, 0;
 0.168, 0, 0, 0.041999999;
 0, 0, 0.041999999, 0]

Energia: 0.146001
Contraste: 0.586
Homogeneidad: 0.8088
Disimilaridad: 0.418
Entropia: 2.09667
GLCM Media: 1.293

```

Figura 11: Medidas de textura obtenidas, para una matriz de co-ocurrencias a 0 grados y distancia 1.

3. Bibliografía

- 1 Using a Gray-Level Co-Occurrence Matrix (GLCM)
<http://tinyurl.com/h8cdco7>
- 2 The GLCM Tutorial, Mryka hall-Beyer
<http://tinyurl.com/yfws wrd>
- 3 P. Mohanaiah, P. Sathyanarayana., 2013, “Image Texture Feature Extraction Using GLCM Approach”, Vol-3, Issue 5.
- 4 Haralick R.M, Shanmugan K and Dinstein I., 1973, “Textural features for image classification”, IEEE Transaction on System, Man and Cybernetics, Vol SMC-3 N.6, pp 610-621.
- 5 M. Presutti, “La matriz de co-ocurrencia en la clasificación multiespectral: tutorial para la enseñanza de medidas texturales en cursos de grado universitario” de 4ta Jornada de Educação em Sensoriamento Remoto no âmbito do Mercosul, São Leopoldo, Brasil, 2004.
- 6 A.B. Girisha, M.C. Chandrashekhar., 2013, “Texture Feature Extraction of Video Frames Using GLCM”, International Journal of Engineering Trends and Technology (IJETT), Vol-4, Issue 6.