

Departamento de Estudios Multidisciplinarios Sede Yuriria

Proyecto Primer Parcial

Visión por Computadora

Elaborado por:

José Baltazar Ramírez Rodríguez

Dra María Susana Ávila García

05 de abril del 2019

1. **INTRODUCCIÓN**

La visión por computadora permite el procesamiento de imágenes para un sinfín de especialidades que van desde la medicina, pasando por la música, hasta el ámbito de seguridad; ya que este procesamiento nos permite mejorar los componentes de una imagen: el contraste, la intensidad de un píxel, la orientación de un píxel, etc. En el área de medicina, por ejemplo, sería fundamental lograr tener una clara percepción de ciertas regiones de una imagen donde se tenga alguna anomalía y poder así determinar un correcto diagnóstico al paciente.

Una de las primeras aplicaciones del procesamiento digital de imágenes, fue en la industria del periódico (Rafael C. González, 1977), cuando las imágenes eran enviadas primero por cable submarino entre Londres y Nueva York. Después se introdujo el sistema de transmisión de imágenes por cable Bartlane, que reducía el tiempo en el que se transportaba una imagen a través del Atlántico de una semana a menos de tres horas. En la actualidad, existen diferentes técnicas y métodos para el procesamiento digital de imágenes.

En este primer proyecto se describirán algunas funciones que fueron implementadas en una interfaz de usuario (GUI) y que permiten operaciones a nivel píxel -siendo este el elemento mínimo que compone una imagen – para modificar y tener mejores resultados en una imagen.

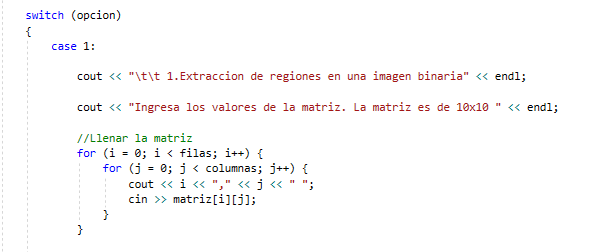
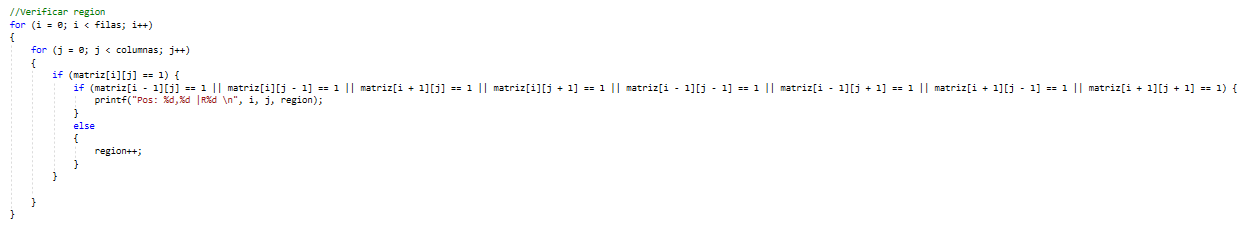
1. **DESCRIPCIÓN DE LA APLICACIÓN**

La aplicación consiste en permitir al usuario, aplicar distintos métodos que logren procesar una imagen. Los métodos que el usuario puede utilizar son:

* Extracción de Regiones en una imagen binaria
* Extracción de bordes de regiones en una imagen binaria
* Transformación Espacial de Traslación
* Transformación Espacial de Escala
* Transformación Lineal de Intensidad de Imagen (Negativo de una imagen)
* Transformación Logarítmica de Intensidad de Imagen
* Transformación de Ley de Potencia de Intensidad de Imagen
* Transformación lineal por partes: Estiramiento de Contraste (Contrast Stretching)
* Transformación lineal por partes: Segmentación de Nivel de Intensidad (Intensitylevel Slicing)
* Cálculo del Histograma de una imagen en escala de grises
* Ecualización de Histograma de una imagen en escala de grises.
* Ajuste de Histograma de una imagen en escala de grises con el de una imagen dereferencia. (Histogram Matching)
* Erosión
* Dilatación

El programa en su función principal tiene un case, que de acuerdo con cada opción desplegará cierto método. El primero es la extracción de regiones en una imagen binaria.

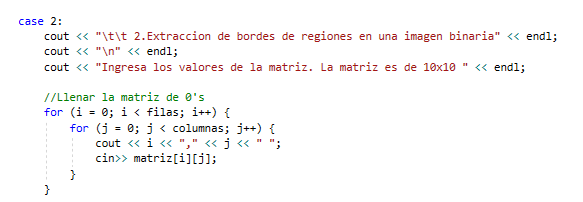
Se comienza solicitando la matriz al usuario. Se define por defecto trabajar con matrices de 10x10.

Después verifica en la matriz en coordenadas i, j = 1 en sus 8 vecinos que tenga estos para saber a qué región pertenece.

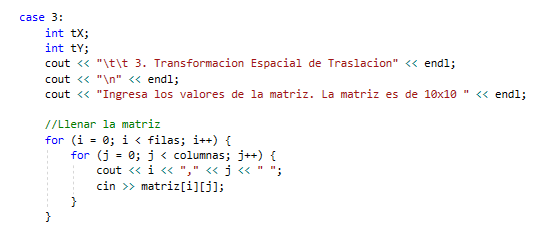
Y se muestra la coordenada y región a la que pertenece.

Respecto al caso 2 del switch, que consiste en extraer bordes de regiones de una imagen binaria, se comienza igual que en la función anterior, solicitando los datos de la matriz de 10x10 al usuario.

Imagen que contiene captura de pantalla

Descripción generada automáticamenteY se verifica si la matriz en sus coordenadas i, j = 1 alguno de los 8 vecinos es igual a cero para determinar que ahí existe un borde e imprime la posición del borde.

Para el caso 3 del switch, la función es la de transformación espacial de traslación. Lo primero que se hace es pedir al usuario los datos de la matriz de 10x10.

Hasta aquí el proceso es el mismo al momento de solicitar los datos en las primeras tres funciones. Luego se piden los factores de traslación en *x* y en *y.* Se visualiza también la matriz original.

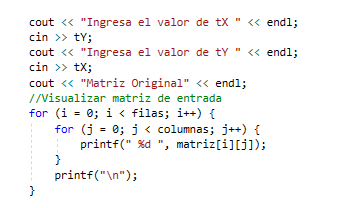
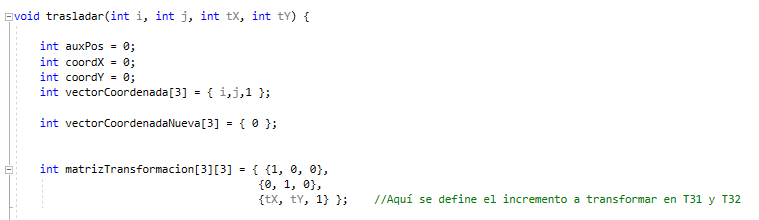


Imagen que contiene persona

Descripción generada automáticamenteLuego se recorre la matriz que se solicita a la entrada y en donde encuentre un ‘1’ en esas coordenadas i,j se manda llamar la función para trasladar.

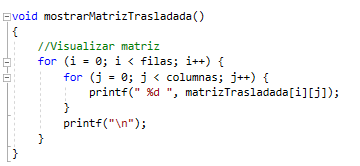
En la función trasladar se tiene la matriz de transformación, el vector con las coordenadas actuales y un vector para almacenar las coordenadas nuevas.



Luego se realiza la multiplicación de la matriz transformación por el vector con las coordenadas actuales. Se utiliza un auxiliar de posición donde acumula la suma de la multiplicación para almacenarse en la primera posición del vector de coordenadas nuevas. Se obtiene de la posición 0 la coordenada x y de la posición 1 del vector de coordenadas. Y en la matrizTrasladada con las posiciones de la coordenada X y coordenada Y se asigna a 1 para poner en esa nueva posición el 1, o sea moverlo en esas coordenadas.

Imagen que contiene captura de pantalla

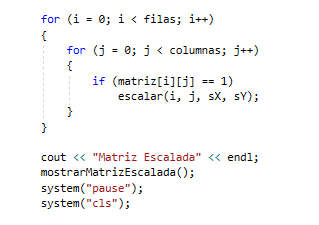
Descripción generada automáticamente

Y se utiliza una función para mostrar la matriz trasladada.

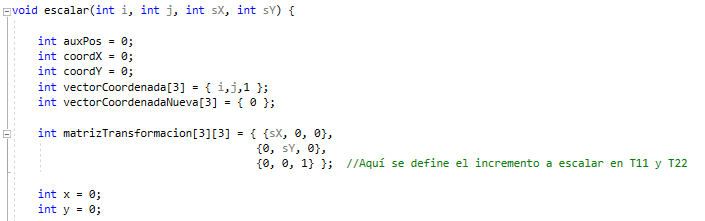
La función 4 para el switch, es la transformación espacial de escala. Se sigue el mismo proceso que en las anteriores opciones. Se solicitan los datos de la matriz.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Se recorre la matriz y donde las coordenadas i,j sean igual a 1, se manda llamar la función de escalar y se pasan los parámetros de escalamiento.

Al igual que para la función trasladar, 𝑎𝑢𝑥𝑃𝑜𝑠 es un acumulador que irá realizando la suma de la multiplicación de 𝑣𝑒𝑐𝑡𝑜𝑟𝐶𝑜𝑜𝑟𝑑𝑒𝑛𝑎𝑑𝑎[𝑖, 𝑗] y la 𝑚𝑎𝑡𝑟𝑖𝑧𝑇𝑟𝑎𝑠𝑛𝑓𝑜𝑟𝑚𝑎𝑐𝑖𝑜𝑛. El 𝑣𝑒𝑐𝑡𝑜𝑟𝐶𝑜𝑜𝑟𝑑𝑒𝑛𝑎𝑑𝑎𝑁𝑢𝑒𝑣𝑎 [3] almacenará nuestro acumulador 𝑎𝑢𝑥𝑃𝑜𝑠 para obtener las nuevas coordenadas de la matriz. Y por supuesto la 𝑚𝑎𝑡𝑟𝑖𝑧𝑇𝑟𝑎𝑠𝑛𝑓𝑜𝑟𝑚𝑎𝑐𝑖𝑜𝑛, que, en este caso, 𝑠𝑥 y 𝑠𝑦 = se solicitan al usuario Además, se declaran dos variables para un ciclo anidado para realizar la multiplicación de 𝑣𝑒𝑐𝑡𝑜𝑟𝐶𝑜𝑜𝑟𝑑𝑒𝑛𝑎𝑑𝑎[𝑖, 𝑗] y la 𝑚𝑎𝑡𝑟𝑖𝑧𝑇𝑟𝑎𝑠𝑛𝑓𝑜𝑟𝑚𝑎𝑐𝑖ón.



Se declara un ciclo anidado para realizar la multiplicación entre los elementos que mencionaron anteriormente y se acumula la suma en 𝑎𝑢𝑥𝑃𝑜𝑠. Se asigna al 𝑣𝑒𝑐𝑡𝑜𝑟𝐶𝑜𝑜𝑟𝑑𝑒𝑛𝑎𝑑𝑎𝑁𝑢𝑒𝑣𝑎[x] la suma de la multiplicación llevada a cabo para obtener la nueva coordenada. Después cuando la matriz original en la posición 𝑖,𝑗 sea = 1, se aplican Imagen que contiene captura de pantalla

Descripción generada automáticamentealgunas condiciones anidadas.

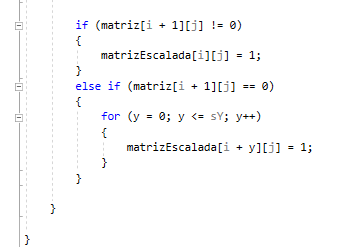
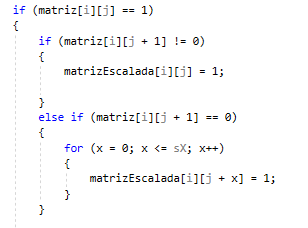
Verifica que la matriz en la siguiente posición en columnas sea diferente de 0, o sea que no

sea un 1, para agregar un 1 en la 𝑚𝑎𝑡𝑟𝑖𝑧𝐸𝑠𝑐𝑎𝑙𝑎𝑑𝑎[𝑖][𝑗]. Con esto garantiza que ponga los

1’s en las mismas posiciones que la matriz original. Si la matriz original en la siguiente

posición de columnas es igual a 0, entonces de define un bucle que vaya de 0 al 𝑠𝑥 para

agregar 1’s a partir de esas posiciones. De esta forma es escala en 𝑥 esa cantidad de veces.

Lo mismo ocurre para las filas. Posteriormente en la función main, se manda llamar a la función 𝑚𝑜𝑠𝑡𝑟𝑎𝑟𝑀𝑎𝑡𝑟𝑖𝑧𝐸𝑠𝑐𝑎𝑙𝑎𝑑𝑎( ) que es de tipo 𝑣𝑜𝑖𝑑.

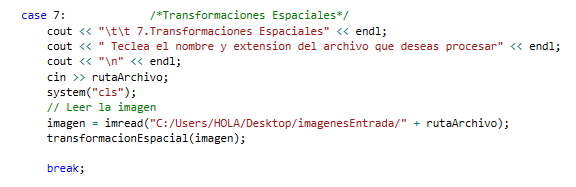
Para la siguiente opción del switch, es la transformación espacial. Se solicita al usuario el nombre de la imagen que desea utilizar, y se manda llamar la función de **transformaciónEspacial** como parámetro la imagen que el usuario eligió.

Imagen que contiene captura de pantalla

Descripción generada automáticamenteSe comienza por obtener su número de filas y columnas de esta imagen de entrada para construir las nuevas matrices de salida, a las que se aplicará la transformación espacial. Estos pasos se muestran en la siguiente figura.

Se decide convertir estas matrices a escala de grises para trabajar con un solo canal y trabajar de una manera más clara y sencilla. La función linear transformation lo que realiza es un inverso de los valores de la imagen (matriz) en cada uno de sus pixeles. Es necesario definir los valores de la ecuación a utilizar que es:

s = L – 1 – r

Imagen que contiene captura de pantalla

Descripción generada automáticamenteDonde r es la intensidad de entrada, s es la intensidad de salida y L-1 es el valor máximo de r.

Se define L como 256 y se declara r. Se recorre la matriz de acuerdo con la cantidad de filas y columnas. Después se define un Scalar llamado intensidad para obtener el valor de cada posición que se va recorriendo con los ciclos anidados. Posteriormente se asigna como r el valor de esta intensidad, la cual será la intensidad de entrada y se realiza la operación. Por último, se asigna a la matriz de grises que había creado el valor de la ecuación y se muestra esta imagen ya con las operaciones realizadas. Se solicita al usuario asignar el nombre de salida del archivo.

La siguiente función es la transformación logarítmica. Se solicita al usuario el valor de la constante c.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

La siguiente función en ser utilizada fue logarithmic transform. Al igual que para la transformación lineal se definió una matriz en escala de grises y se operó sobre ella. Se utiliza una c que se pide al usuario y simplemente se cambia la fórmula en la función. Y se muestra la salida. La fórmula a seguir es 𝑠 = 𝑐 ∗ log(1 + 𝑟). Se verifica que el valor de intensidad de salida no sea mayor que el rango en la escala de grises. De ser así, se asigna un máximo de 255 para aproximarlo a un color blanco (255). Se solicita al usuario asignar el nombre de salida de la imagen.

La última operación fue power law transform. Los valores que se definen para la ecuación

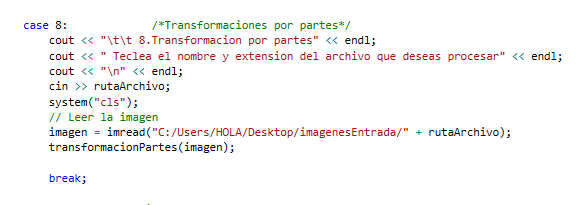
𝑠 = 𝑐 ∗ 𝑟

𝛽 donde c es una cte y beta otra constante. Los primeros valores que se definen son c y beta que ingresa el usuario. En el código beta fue nombrado gamma. Se sigue el mismo comportamiento que en lo anterior. Definiendo una matriz para asignar los nuevos valores. Se verifica que el valor de intensidad de salida no sea mayor que el rango en la escala de grises. De ser así, se asigna un máximo de 255 para aproximarlo a un color blanco (255).

Imagen que contiene captura de pantalla

Descripción generada automáticamente

La siguiente opción en el menú es la transformación por partes. Se solicita el nombre del archivo y se manda llamar la función transformación por partes como parámetro la imagen.



Se comienza por obtener su número de filas y columnas de esta imagen de entrada para construir las nuevas matrices de salida, a las que se aplicará la transformación espacial. Estos pasos se muestran en la siguiente figura.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Se solicitan los valores al usuario mediante consola

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Se decide convertir estas matrices a escala de grises para trabajar con un solo canal y trabajar de una manera más clara y sencilla. El contrast stretching es un proceso que expande el rango de niveles de intensidad en una imagen.

Posteriormente se recorre la matriz de acuerdo con sus filas y columnas, y se define un Scalar para obtener la intensidad de cada píxel. Se extrae la intensidad de cada píxel y se asigna a 𝑟. Lo siguiente es comprobar una de las tres condiciones para definir qué fórmula aplicar para obtener la pendiente de acuerdo con la intensidad de r.

Imagen que contiene captura de pantalla, texto

Descripción generada automáticamente

Y se solicita al usuario que ingrese el nombre del archivo de salida.

La siguiente función es la de intensity-level slicing. Se solicita al usuario los valores del rango para la intensidad.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Y se solicita el nombre de salida del archivo y se muestran las imágenes.

Lo siguiente en el switch es el histograma que recibe como parámetro la imagen de entrada. Se solicita al usuario el nombre de dicha imagen y se mandar llamar la función histograma.

Imagen que contiene captura de pantalla

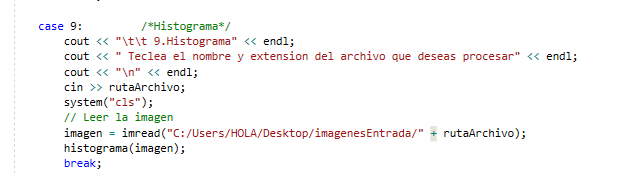
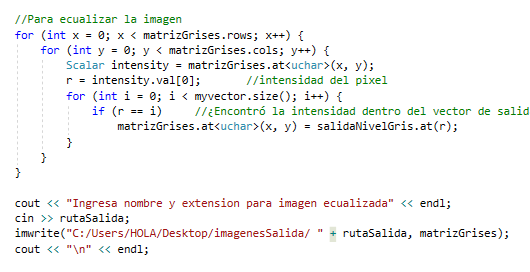
Descripción generada automáticamenteEn la función de histograma igual que en el resto, se verifica que se cargue la imagen y se obtiene filas y columnas de la matriz. Se crean 2 matrices en escala de grises para comenzar a operar. Se declaran las variables necesarias para calcular el histograma así como un vector para guardar las ocurrencias.

Imagen que contiene interior

Descripción generada automáticamenteSe calculan las ocurrencias de intensidad de gris de la imagen.

Para calcular el histograma se siguen las fórmulas vistas en clase y se abre un archivo para comenzar a guardar estos valores en un archivo .csv. El valor de salida se almacena en un vector de 256 posiciones para luego asignar este a la imagen ya ecualizada.

Imagen que contiene captura de pantalla

Descripción generada automáticamenteLuego se recorre la matriz y se asigna el nuevo valor de salida al nivel de intensidad que corresponda. De esta forma queda ecualizada la imagen. Se pide al usuario nombre de salida de la imagen ecualizada.

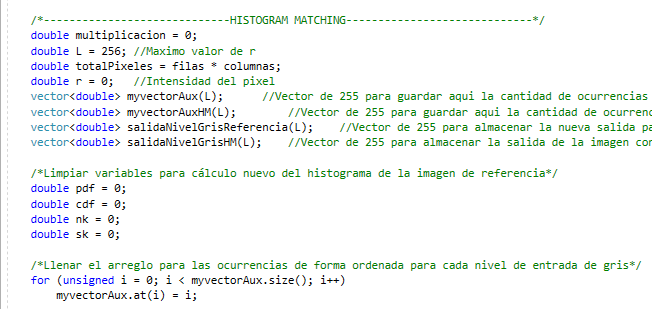
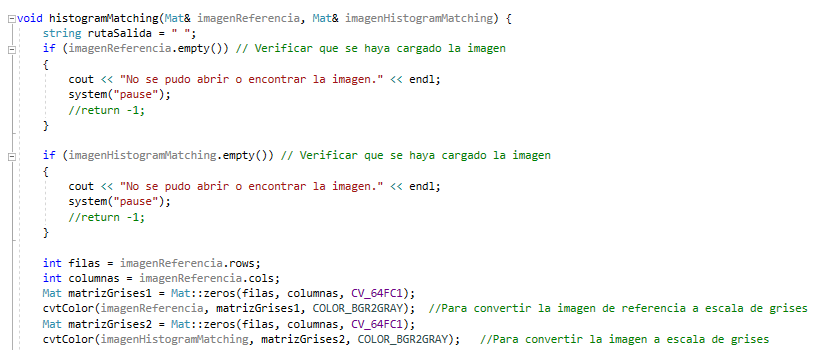
La siguiente función es la de Histogram Matching. Que repite el mismo proceso que la del histograma, sólo que ahora el nivel de salida de la imagen de referencia, será el nivel de entrada de la imagen que hay que ecualizar y se calcula de nueva forma su histograma. Por lo que las imágenes se pondrán secuencialmente. Igual se guarda un archivo con los valores de histograma y se solicita el nombre de las imágenes de referencia, a la que hay que aplicar el histogram matching y el nombre de salida de la imagen ya ajustada.

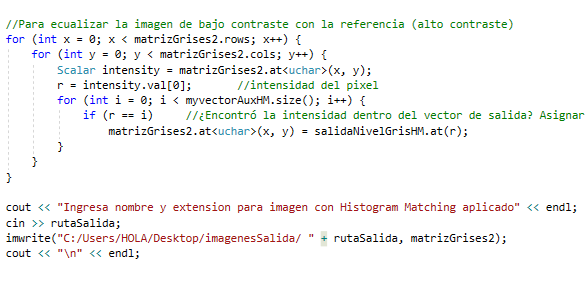
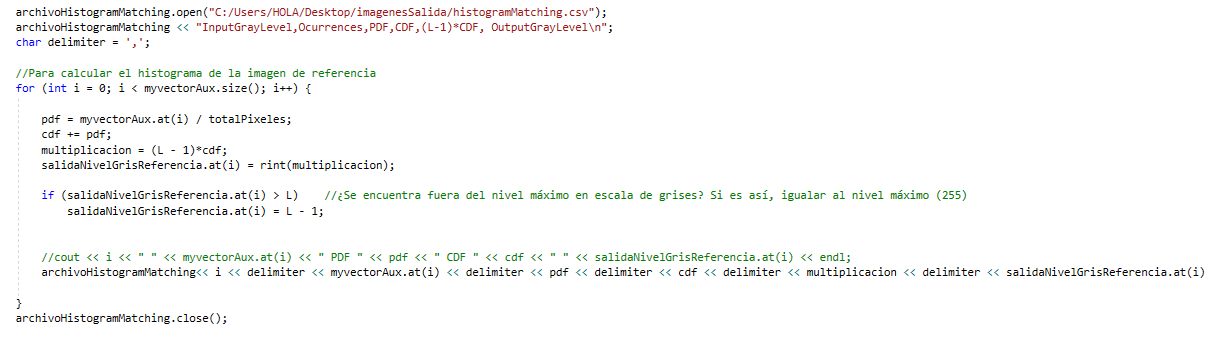
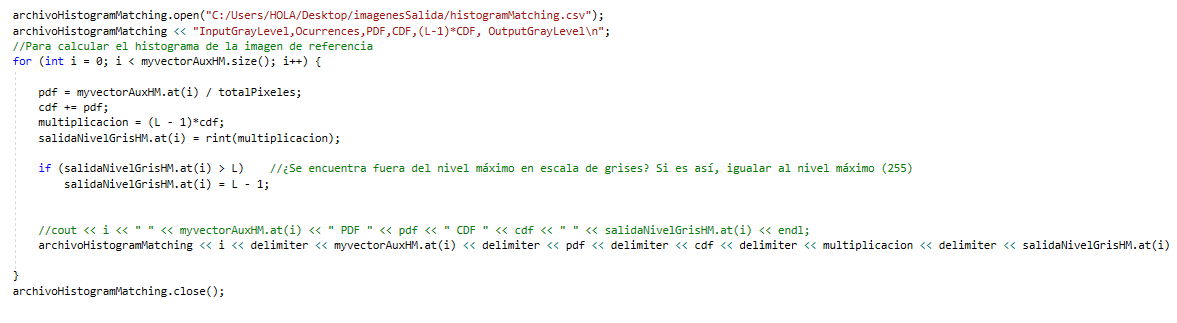
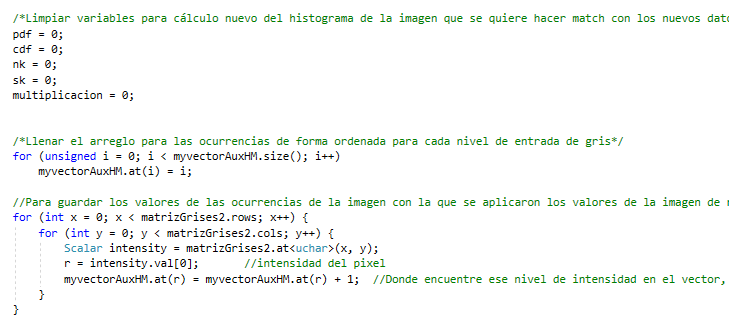
También se solicita una vez que entra al case las imágenes para poder comenzar a operar. El código se muestra a continuación.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

Luego se pasan como parámetro ambas imágenes y se comienza a realizar el mismo proceso que en el histograma.

Imagen que contiene captura de pantalla

Descripción generada automáticamente

1. **ANÁLISIS DE LOS RESULTADOS**

Los resultados que se obtuvieron de las imágenes que se muestran en las pruebas son bastante buenos. Por ejemplo, en el caso de histograma la imagen original, en escala de grises se percibe muy oscura y sufre un cambio en contraste. Se observa ya más iluminada y al recibir esta como entrada al hacer match con el histograma actual, esta se percibe aún más clara.

Para el caso de las transformaciones por partes, el constrast stretching resulta bastante eficiente ya que, al ampliar el rango de forma dinámica, se logran visualizar las nubes y las montañas de una forma más clara. Evidentemente esto trajo consigo una mejora en la calidad de la imagen de entrada. Se puede observar si se compara con la imagen original, que el contraste aumentó y que se distingue de mejor manera el cielo de la imagen, las montañas y las nubes en la parte izquierda de la imagen también pueden apreciarse mejor. Aunque existe una zona en la imagen que está muy oscura que es casi en la parte central, debajo de las montañas. Definitivamente este es la mejor opción para mejorar esta imagen, ya que el constrast stretching se dice que normaliza la imagen, porque “estira” el rango de valores de intensidad que contiene para abarcar un rango de valores deseado. Los puntos (r1, s1) y (r2, s2) controlan la forma del contraste.

Y por último para el primer caso de las transformaciones espaciales, aunque logran su objetivo, se considera que con un caso de estudio de rayos x, o médico se pudiera aplicar de mejor forma estos algoritmos de transformaciones espaciales. El mejor de estos tres procesos, de forma personal, es la transformación lineal.

1. **REFERENCIAS**

[1] Rafael C. González-Richard E. Woods, 1977, “Digital Image Processing”, Pearson-Prentice Hall.