EIE

Escuela de
Ingeniería Eléctrica

UNIVERSIDAD DE COSTA RICA

UNIVERSIDAD DE COSTA RICA

FACULTAD DE INGENIERIA

ESCUELA DE INGENIERIA ELÉCTRICA

ESTRUCTURAS ABSTRACTAS DE DATOS Y ALGORITMOS PARA INGENIERÍA

PROPUESTA PROYECTO 0: EFECTOS EN IMAGENES DIGITALES

ESTUDIANTES:
JESÚS ZÚÑIGA MÉNDEZ
DENNIS CHAVARRÍA SOTO

PROFESOR:
RICARDO ROMÁN BRENES; M. SC.

II CICLO 2019

Índice

1. Reseña del programa

Desde la invención de la fotografía, han existido diferentes formas para editar las imágenes y añadir efectos que las alteran, otorgándoles un estilo muy diferente; del mismo modo, durante el siglo XX era frecuente que muchas imágenes de colectivos se editaran, fuese para ocultar personas de las fotografías, fuese para crear diseños psicodélicos, entre muchas otras razones (Murias, 2017). Con ello, los métodos y las posibilidades han evolucionado de forma impresionante, al punto de existir aplicaciones que permiten utilizar filtros de color, añadir elementos a rostros de las personas, e incluso, crear variaciones en los rasgos faciales en un retrato; una diversidad de posibilidades que son posibles gracias a la programación de un código que permite su funcionamiento.

El proyecto número 0 del curso de Estructuras Abstractas de Datos y Algoritmos para Ingeniería, tiene como finalidad el desarrollo e implementación de un programa que sea capaz de aplicar filtros y operadores morfológicos a una imagen que el usuario especifica, así como cambiar el tipo de formato de la misma y regresar la respectiva copia con las modificaciones realizadas. Específicamente, el formato de salida, así como las operaciones que deben aplicarse, son señaladas cuando se utiliza el programa.

Como se mencionó anteriormente, la edición fotográfica juega un papel importante en la sociedad, sea como ocio o como una profesión; por lo que el desarrollo de este proyecto tiene gran valor en cuanto se combina la aplicación de las habilidades de programación aprendidas en el presente curso, así como los anteriores y se desarrolla un programa que introduce a los estudiantes a la creación de software que puede resultar útil en la actualidad. Además, el lenguaje de programación que se utilizará para desarrollar el código es C++.

2. Funcionamiento del programa

El programa requiere de un grupo de funciones especializadas que modificarán una imagen. Primeramente, las operaciones disponibles son: (FG) Filtro gaussiano; (FSTD) Filtro de desviación estándar, (ED) Detección de bordes -edge detection-, (MB) Difuminado de movimiento (motion blur), (S&P) Ruido sal y pimienta, (E) Erosión, (D), dilatación, (I) Inversión de color, y (G) Transformación de escala de grises. Tales funcionalidades se invocan mediante la línea de comandos, donde se utiliza la sintaxis ./proyecto0 IMAGEN.FORMATO OPERACION NUEVOFORMATO. la primera parte corresponde al nombre del archivo ejecutable, luego, IMAGEN.FORMATO es el nombre de la fotografía que se desea editar; la operación se elige mediante las abreviaciones entre paréntesis anteriormente mencionadas; por último, NUEVOFORMATO se utiliza para indicar el formato final que tendrá la imagen modificada. Cuando se invoca el programa, y edita la imagen que el usuario indica, el resultado recibe el nombre IMAGEN.EXTENSION.OPERACION.NUEVOFORMATO. Además, es necesario recurrir a librerías de terceros para procesar las imágenes, leerlas y escribirlas.

2.1. Filtro Gaussiano

Este filtro se utiliza para hacer que una imagen se vea borrosa, es decir, desenfoca parcialmente parte de la imagen. Es utilizado para suavizar imágenes, reducir el ruido en ellas. El procedimiento consiste en aplicar a cada pixel de la imagen el llamado filtro, que, basado en una matriz, varía una serie de valores en los pixeles, limitando los datos que se permiten para reescribirlo y lograr el efecto de desenfoque; como se trata de la variante Gaussiana, el valor máximo de filtrado se localiza en un pixel central, para luego disminuir conforme se aleja de este (universidad de Murcia, s.f.).

2.2. Filtro de desviación estándar

Este filtro, muy similar al Gaussiano, aplica el procedimiento de la matriz a cada pixel, generalmente de 3x3, para determinar la desviación estándar entre ellos y filtra los datos de la imagen que no están comprendidos en tal intervalo. Nuevamente, el programa utiliza el filtro para crear un efecto de desenfoque (Mathworks, s.f.-b).

2.3. Detección de bordes (Edge Detection)

La detección de bordes es utilizada por el programa para procesar imágenes y lograr la detección de discontinuidades o segmentación -Procesos muy importantes en la robótica, por ejemplo, o la visión computador- (Mathworks, s.f.-a). El procedimiento consiste en tomar la imagen, y, mediante unas funciones, detectar variaciones en el brillo para determinar la distinción de los bordes de los cuerpos en las fotografías. Se puede basar en la teoría del algoritmo de Canny; además utiliza las herramientas de matrices tal y como el Filtro Gaussiano, gracias a las posibilidades proporcionadas por OpenCV (2019).

2.4. Difuminado de movimiento (Motion Blur)

Para el Motion Blur, la aplicación debe generar un desenfoque que se realiza en imágenes cuyo fondo está en movimiento respecto a un marco inercial anclado a un cuerpo. Al programar este efecto se puede utilizar una combinación del filtro gaussiano y la detección de bordes, de forma que lo que es exterior a determinados objetos, los cuáles se supone no están en movimiento, será distorsionado por el programa.

2.5. Ruido sal y pimienta (S&P)

El filtro sal y pimienta agrega ruido a la imagen, es decir hace que la imagen quede con una especie de textura que asemeja sal o arena, por eso su nombre, esto se logra agregando pixeles con colores blanco y negro a toda la imagen, la dispersión de dichos pixeles depende la intensidad del que se desee aplicar.

2.6. Erosión

EN el filtro de erosión se quitan las capas mas altas de una imagen, esto al decirlo es abstracto pero en la practica es un poco mas comprensible, básicamente se define un elemento estructurador(un pixel), y si en el vecindario de este elemento, es decir en los pixele vecinos este elemento es el mas alto entonces se erosiona, es decir se le baja su valor.

2.7. Dilatación

El filtro de Dilatación es justamente lo contrario de el de Erosión, en este caso se toma un elemento estructurador menor y el vecindario de este es el que se dilata, es decir se agranda.

2.8. Inversión de color

En el caso de la inversión de color el filtro realiza justamente lo que su nombre indica, si partimos de el echo que en una computadora todo se interpreta como números, en el caso de este filtro es cambiar el valor de un pixel justo al valor que corresponde contrariamente, es decir en una distribución de 256 colores, si tenemos un numero por ejemplo 128 su contrario seria el 256.

2.9. Transformación escala de grises

La escala de grises se logra promediando cada pixel en cada una de las matrices que corresponde al RGB, al realizar este promedio se le asigna un tono de gris dependiendo del valor que se obtenga, este filtro tiene la necesidad de manipular de una forma muy exacta las matrices para que el resultado obtenido sea el esperado.

3. Experimentos

Para la prueba y validación del programa que se realizará se harán algunos experimentos con cada uno de los filtros que se programarán, la plataforma escogida es el lenguaje de programación C++ junto al sistema operativo Linux, se hará un Script en Bash que facilite la corridas del programa para cada filtro con la misma imagen y con imágenes distintas, para las diferentes etapas de pruebas se espera lo siguiente:

1. Se realizarán corridas con cada filtro con imágenes cuyo tamaño no sea muy grande, esto para que cada corrida sea mas rápida
2. En cada corrida se inspeccionara la imagen de forma visual contrastándola contra la imagen original con el objetivo de juzgar la calidad del filtro implementado.
3. Una vez el filtro este optimizado se realizarán corridas con imágenes mas grandes con el fin de determinar el comportamiento de cada filtro con la variación de tamaños.
4. Después de validar cada filtro con distintos tamaños se optimizara el código en busca de mejoras para la ejecución.

se espera que las imágenes filtradas por el programa se puedan apreciar de forma contundente al contrastarlas contra las imágenes originales, después de haber logrado cada filtro se espera poder medir el tiempo de ejecución de cada corrida para buscar alguna forma de optimizar el tiempo de ejecución del programa.

4. Bibliografía

Cortés, E. (2011). *La máquina universal.* Recuperado de: http://www.programando.org/aprende-a-programar/segunda-parte/arquitectura-de-computadores/la-maquina-universal.html.

Mathworks. (s.f.-a). *Edge detection.* Recuperado de: https://www.mathworks.com/discovery/edge-detection.html.

Mathworks. (s.f.-b). *stdfilt.* Recuperado de: https://es.mathworks.com/help/images/ref/stdfilt.html.

Murias, D. (2017). *16 fotos manipuladas que pasaron a la historia.* Recuperado de: https://magnet.xataka.com/un-mundo-fascinante/16-fotos-manipuladas-que-pasaron-a-la-historia.

OpenCV. (2019). *Canny edge detector.* Recuperado de: https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html.

Platero, C. (s.f.). *Procesamiento morfológico.* Recuperado de: http://www.elai.upm.es/webantigua/spain/Asignaturas/Robotica/ApuntesVA/cap6VAProcMorf.pdf. Universidad Politecnica de MAdrid.

universidad de Murcia. (s.f.). *Técnicas de filtrado.* Recuperado de: https://www.um.es/geograf/sigmur/teledet/tema06.pdf. Universidad de Murcia.