Visión por Computadora
Práctica 1 (reporte)

Jonathan de Jesús Andrade López

10 de marzo de 2015
Facultad de Ciencias UNAM

Detalles de la implementación, problemas y momentos de diversion en la práctica 1.

Objetivo: Mejorar la intensidad de color y luminiscencia, además de aplicar fltrado en imágenes mediante correlación y convolución

Introducción: Un punto clave de este curso es la extracción de información de una imágen, para ello, es preciso tratarla antes de sacarle el mayor provecho a nuestros pixeles. Parte de ese proceso de tratamiento incluye mejorar su color, cambiar sus dimensiones, usar otro sistema de coordenadas, suavizarla, para mejorar las carácteristicas que realmente nos pueden servir a procesar correctamente la imágen. Estos filtros o procedimientos son básicos, forman parte de algoritmos más complejos que posteriormente revisaremos.

Los filtros de **Desenfoque gaussiano** y **Laplaciano del gaussiano**, se calculan usando matrices de convolución, i.e iterando sobre los pixeles de la imagen, tomaremos una matriz con un tamaño en relación al valor de la intensidad con la que queremos aplicar el filtro. Esta matriz, contiene los *pesos* relativos a la vecindad del pixel, y asi ponderar su importancia relativa al pixel angular del filtro en ese indice de la iteración.

El resto de los filtros son lineales, pues solo se recorre la imágen una unica vez y esto basta para obtener el resultado requerido.

Desarrollo: Los múltiples filtros que se implementaron en esta práctica son base para mejorar las condiciones con las que recibimos una imágen, pueden ayudar a mejorar su color, brillo, contraste, suavizarla, en otras palabras, preparla para la extracción de información y manipulación más específica dependiendo del propósito de nuestro procesamiento.

- **Balance de color:** Se usa para mejorar la intensidad del color de las imágenes en base al muestreo de intensidades de color de cada canal en el espectro RGB. Tomando el valor máximo y el mínimo en el intervalo [0, 255], y haciendo la regla de correspondencia de estos valores hacia el cero absoluto y el máximo absoluto dentro del intervalo. En imágenes con una gama baja y constante de colores, el filtro tiene resultados visibles, pero, cuando al menos un pixel esta en alguna de los límites del intervalo, su efecto es nulo. Como solución a este problema, en lugar de tomar los valores extremos del espectro de intensidades, se puede tomar la media de cada canal, y hacer un proceso similar en dos direcciones, de la media hacia el cero, y de la media hacia el 255. Sus resultados son más visibles cuando se aplica después de aumentar el brillo del canal de la imágen (desplazarla en la escala)
- **Blending:** Este filtro es de los más sencillos, pues requiere de dos imágenes mezclando los pixeles de su área de intersección bajo la siguiente regla de correspondencia:

  $$C = (1 - \alpha)B + \alpha F$$

  Con B y F las imágenes de fondo y de primer plano respectivamente.
- **Ajuste de contraste:** Este filtro tiene mayor presencia en la imágen después de aplicarse, mejora el ajuste general de la imagen, pues no solo hace una correspondencia lineal entre las magnitudes de de los colores de cada canal, si no que toma en cuenta la *función de probabilidad acumulada* es decir distribuye uniformemente la intensidad de los colores mapeando las tonalidades anteriores a unas nuevas en función de su distribución.
- **Desenfoque Gaussiano:** Uno de los filtros más importantes dentro del procesamiento de imágenes, pues permite que cada pixel tome "información" de sus vecinos, ponderando su imporntacia en

función de su distancia, es decir, los vecinos más cercanos, tendrán más peso al momento de dejar su parte en el pixel de atención. Aplicar este filtro, requiere de una matriz de convolución, es decir, se iterara sobre las dimensiones de la imágen, pero además se recorrera linealmente la matriz (de tamaño $2\sigma + 1$) para poder pesar sus pixeles vecinos. Una ventaja de este algoritmo es que su kernel (o matriz de convolución) es separable, es decir, se puede aplicar primero en sentido horizontal, y después en vertical obteniendo el mismo resultado, esta propiedad se debe a que a función gaussiana es simetrica y radial. Esta propiedad es preciosa en sentido computacional, pues reduce la complejidad del algoritmo.

$$O(wh4\sigma^2) \longrightarrow O(wh4\sigma)$$

Donde w, h son el ancho y alto de la imagen, y σ el tamaño del kernel, en términos de tiempo, con una imagen de 12 Megapíxeles, el tiempo se reduce de $40.95w\ ms$ a tan solo $9.8\ ms$ (Java 8 64 bit GNU/Linux)

- **Laplaciono del Gaussiano (Aproximación):** Como vimos en clase, el operador Laplaciano es la regla de la cadena de doble derivada de la funcion gaussiana, viendo el resultado geométrico, podemos notar que su resultado es muy similar al de aplicar el filtro Gaussiano a la imágen y restar estos valores a sus intensidades originales, de esta manera, obtenemos solo aquellos colores de gama alta, dado que sólo es una aproximación, el resultado es bueno, pero no el más preciso, pero por fines pragmáticos, este es el método más utilizado.
- **Imágenes Hibridas:** En este filtro usamos los dos anteriores, primero, tomamos una imágen, aplicamos el desenfoque Gaussiano para eliminar las frecuencias más altas, después, tomamos una segunda imágen y a esta le aplicacimos el laplaciano, de esta manera nos quedamos las frecuencias más altas de una y las más altas de la otra, dando un efecto visual, que se percibe a diferentes distancias. Cabe mencionar que el valor de la σ es independiente para cada filtro, de esta manera, obtendremos el resultado deseado, y como es evidente, la forma de logra el efecto, es mezclando las imagenes usando el blending en una proporción de $\alpha = 0.5$

- **Mundo pequeño:** En cuestion de implementación, este fue el filtro que más tarde en solucionar. Consiste en tomar una imagen panoramica, rotarla, hacerla cuadrada y por último, aplicar una transformacion polar para verla como un círculo. Uno de los problemas que más tarde en resolver fue el de mapear la imagen cartesiana a formato polar, primero intente iterar sobre un círculo sobre el diámetro y el valor de su radio, pero como consecuencia, muchos de los puntos no estaban definidos bajo esta función, así que como nos habian mencionado en clase, el truco estaba en iterar sobre la imágen resultado, calculando a cada pixel el valor de r y de Θ, una vez que los tenemos, solo basta considerar, que el ancho de la imagen equivale a 2π y el alto al valor esta en relación con r y así obtenemos el relleno total de los pixeles destino de la imágen.
- **Interpolación** Sirve para poder escalar (UpScale, DownScale) una imágen, usando la formula matemática de la interpolacíon bilienal, usando los pixeles aledaños para calcular el valor del pixel, que esta entre ellos (en caso de que se se este aumentado el tamaño). Es u proceso relativamente rápido y con buenos resultados, aunque en mi caso, aun tengo problemas con algunos colores.

Conclusión: Los diferentes filtros que se implementaron en esta práctica, son lo fundamental para poder desarrollar procedimientos más complejos con el tratamiento de imágenes, el obejtivo fundamental de tratar las imágenes es dejarlas en las mejores condiciones para extraer de ellas las carácteristicas que nos pueden dar más información de la imágen. Mejorar el color, el contraste, cambiar el tamaño, suavizar y cambiar el sistema de coordenadas, pueden ser de gran utilidad para futuras aplicaciones dentro del curso, así que por esta práctica, considero cumplido el objetivo.