VC: Informe práctica 1

Gustavo Rivas Gervilla

Vamos a dividir el informe en distintas secciones dedicadas a explicar lo que hemos hecho para completar cada una de las tareas que contenía esta práctica:

A. La convolución.

calcularVectorMascara:

Lo primero que tenemos que hacer es crear la máscara. Vamos a aprovechar que estamos muestreando una Gaussiana que como sabemos es separable y simétrica. Por lo tanto lo que vamos a construir va a ser una máscara 1D que utilizaremos tanto para las filas como para las columnas.

El tamaño de la máscara vendrá en función del σ y será el siguiente: $round(3\sigma)2 + 1$ esto se debe a que queremos muestrear dentro del intervalo $[-3\sigma, 3\sigma]$ y ponemos así el redondeo para poder tener máscaras impares de cualquier tamaño. Si por ejemplo hiciésemos $3ceil(\sigma)2 + 1$ el tamaño mínimo de las máscaras sería de 7. Con lo cual hemos de hacer el cálculo como hemos dicho. Como vemos lo hacemos de modo que obtengamos máscaras de orden impar.

El siguiente cálculo importante que tenemos que hacer es el del paso de muestreo, pretendemos que el mayor peso recaiga sobre el píxel central con lo cual el tamaño del paso será: $\frac{6\sigma}{longitud - 1}$ y así también aseguramos muestrear en los extremos del intervalo relevante que hemos señalado antes.

Ya lo único que queda es muestrear la Gaussiana de párametro σ y dividir cada componente por la suma de todos para que la máscara sume uno, una condición de los filtros de alisamiento.

obtenerVectorOrlado:

Como sabemos la convolución, al estar trabajando con un núcleo sepable, se realiza en filas y en columnas por separado. Pues bien lo que hace esta función es tomar una matriz 1D y prepararla para realizar sobre ella la convolución. Es decir, la orlamos añadiendo a sus extremos tantos píxeles como sean necesarios para poder convolucionar correctamente.

En concreto si nos situamos en los extremos de la fila/columna a convolucionar sobrarán los píxeles de la máscara a uno de los lados del píxel central, con lo cual la cantidad de píxeles a añadir es: mascara.cols - 1.

Una vez orlada la fila/columna a convolucionar con esta cantidad de píxeles (la mitad a cada lado) no tenemos más que rellenarlos o bien poniéndolos a cero o en modo espejo, según elijamos con los parámetros.

calcularConvolucionVectores1C:

Una herramienta muy útil son las funciones *split* y *merge* de OpenCV que nos

permiten obtener por separado cada uno de los canales de la imagen y luego juntar varios canales en una imagen (respectivamente). Con lo cual con el objetivo de reutilizar código y trabajar de la forma más estándar posible lo que haremos es escribir código para imágenes con un sólo canal, luego si tenemos una imagen con 3 canales en la función genérica los separamos, procesamos cada uno por separado (con la versión para un sólo canal) y los volvemos a unir. Por eso de ahora en adelante sólo nos centraremos en explicar el funcionamiento de las versiones 1C de las funciones.

En esta función lo único que hacemos es aplicar la operación de convolución a una matriz 1D. Teniendo en cuenta que es un vector orlado y con lo cual empezamos a hacer la convolución sólo en los píxeles que no son de la orla y aprovechando la función colRange para fijar un ROI en cada paso.

Observemos que la máscara está preparada para trabajar con vectores fila, esto es así porque simplemente con trasponer ya podemos trabajar con las columnas como si fuesen filas.

convolucion2D1C:

Aquí simplemente aplicamos la convolución a una imagen, primero a las filas y luego a las columnas. Con lo cual lo que hacemos es convolucionar las filas, trasponemos, convolucionamos las columnas (como si fuesen filas) y deshacemos la trasposición.

[Ejemplo en el que se vea como se defumina más la imagen cuanto mayor es el sigma]

Como podemos ver cuánto más grande es el sigma, y sobre todo cuando se aumenta el orden de la máscara la imagen se difumina más, esto se debe a que cada vez influyen más píxeles (cada vez más alejados del central) en el valor de uno.

B. Imágenes híbridas

El código es muy sencillo; como sabemos para obtener una imagen híbrida lo único que tenemos que hacer es obtener las frecuencias bajas de una imagen (aplicar un filtro de alisamiento, con las funciones de convolución de la parte A) y obtener las altas de una imagen, restándole a la original sus frecuencias bajas.

[Ejemplo del perro-gato]

Para cada pareja de imágenes habrá que ajustar adecuadamente los sigmas de modo que el efecto sea el mejor posible.
[Reflexión sobre sigmas del gato perro]

C. Pirámide Gaussiana

calcularPirGaussiana1C:

Aquí lo único que hacemos es aplicar el mecanismo básico para obtener las pirámides Gaussianas. Partimos de la imagen original como primer nivel, entonces para obtener un nivel lo que hacemos es alisar el anterior y submuestrearlo, esto es tomamos sólo las columnas y filas impares.

Lo que hay que tener en cuenta es el sigma para el filtro de alisamiento, dado que vamos saltanto las columnas pares no tiene sentido tener en cuenta lo que pase a más allás de un píxel de distancia del central. Por lo tanto vamos a tomar una máscara de tamaño tres y en consecuencia por los cálculos que hemos descrito anteriormente, tomamos un σ menor que 1, mayor en menor en función del peso que queramos darle a cada uno de los píxeles que intervendrán en la convolución.

[Ejemplo piramide gato-perro]
Aquí podemos ver el efecto que se busca en las imágenes híbridas.