Master: Ciencia de Datos e Ingeniería de Computadores

Curso: Visión por Computador

Rosa Mª Rodríguez Sánchez

Dpto. Ciencias de la Computación e I. A.

E.T.S. de Ingenierías Informática y de Telecomunicaciones

Universidad de Granada

Filtrado espacial

Índice de contenido

1. Introducción

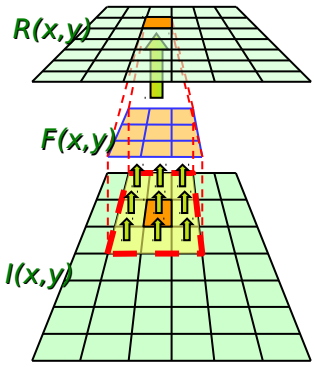
Las operaciones de filtrado espacial consisten en, dado un punto de una imagen, transformarlo de acuerdo a su valor y al de un cierto grupo de vecinos. La misma operación se hace en todos y cada uno de los puntos de la imagen que estamos transformando. Cuando la operación que hacemos con los vecinos es una transformación lineal, entonces hablamos de filtrado lineal espacial.

Las operaciones más simples y más utilizadas de este tipo con la correlación y la convolución.

2. Correlación

Dadas dos imágenes I(x,y) y F(x,y), la correlación entre ambas, R(x,y), se calcula mediante la expresión:

$$R(x,y)=\sum_{i=-N}^{N}\sum_{j=-N}^{N} I(x-i,y-j)*F(i,j)$$



donde 2N+1 es el tamaño de la imagen F. Por simplificar la escritura de la expresión anterior, F(0,0) se corresponde con el centro de la imagen (y no con la esquina superior izquierda como suele ser habitual).

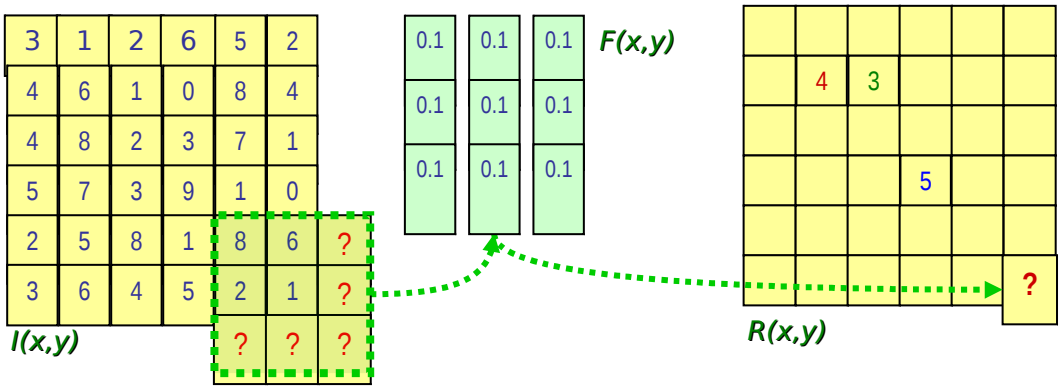
F es conocida como filtro o núcleo de correlación y suele tener tamaño cuadrado e impar.

En Matlab disponemos de la función imfilter para calcular la correlación entre imágenes:

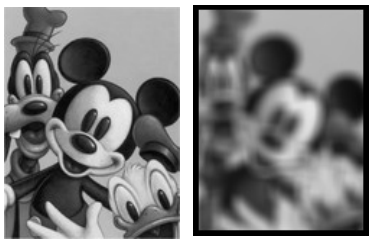
```
>> resultado = imfilter(img,filtro);
```

Devuelve una imagen que se obtiene aplicando la expresión anterior a cada uno de los puntos de la imagen img. La imagen de entrada puede ser de cualquier tipo y el resultado es del mismo tipo que ella. Si la imagen de entrada es de tipo entero entonces el resultado también lo es, perdiendo los posibles decimales o truncando el número si excede los límites de representación de los enteros.

Existen algunos puntos para los que, en principio, no está definida esta operación: los bordes de la imagen. Por ejemplo, en la figura inferior vemos una imagen I(x,y) y un filtro F(x,y). Al aplicar el filtro sobre el pixel de la esquina inferior derecha, nos faltan valores en I(x,y).



Al no estar definida la correlación en esos puntos, el resultado sería una imagen un poco más pequeña, o bien del mismo tamaño pero con un valor arbitrario en esos puntos (por ejemplo cero). A la derecha vemos una imagen y el resultado de filtrarla con un núcleo de tamaño 9x9. Vemos como alrededor hay un borde de 4 píxeles en el que no ha sido posible hacer ningún cálculo.



Para evitar ese efecto, o al menos minimizarlo, se suelen adoptar distintas estrategias. Por ejemplo, considerar que los puntos en donde I(x,y) no está definida valen un valor constante (por ejemplo cero). También es frecuente considerar que la imagen es toroidal, es decir, que la columna que sigue a la columna más a la derecha es la primera de la izquierda y viceversa y de forma análoga se hace con las filas.

La función imfilter dispone de un parámetro para indicar estas consideraciones. Este parámetro puede valer:

- Un valor constante. Cualquier punto en donde I(x,y) no esté definida valdrá este valor.
- 'symmetric'. Se considera que la imagen es simétrica en cualquiera de sus extremos.
- 'replicate'. Los puntos inexistentes toman el valor del pixel más cercano.
- 'circular'. Se considera que la imagen se repite de forma periódica.

```
>> resultado1 = imfilter(img,filtro,128);
>> resultado2 = imfilter(img,filtro,'symmetric');
```

En este ejemplo, el primer filtrado supone que los píxeles no definidos valen 128. En el segundo caso, se supone que la imagen es toroidal, es decir, simétrica a partir de cualquiera de sus bordes. Observa que esta última situación evita transiciones bruscas en los bordes de la imagen.

Original *Constante=0* *Constante=128* *'circular'* *symmetric'*

3. Filtros predefinidos en Matlab

Para crear algunos filtros de uso bastante frecuente disponemos de la función fspecial. En concreto podemos crear los siguientes:

- 'average'. Crea un filtro para calcular la media de los puntos del entorno de un pixel dado. El tamaño del entorno es un parámetro adicional.

  ```
  >> f1 = fspecial('average',7);
  >> f2 = fspecial('average',[7 15]);
  ```

  En este ejemplo f1 es una máscara cuadrada de 7x7 tal que cada uno de sus coeficientes vale 1/49. De esta forma, al calcular la correlación lo que estamos haciendo es calcular la media de todos los valores del entorno 7x7 del punto en cuestión. f2 es una máscara que define un entorno de 7x15 (rectangular) sobre el que se calculará la media.

- 'disk'. Crea un entorno circular para calcular la media del entorno del píxel. Necesita como parámetro el radio del círculo.

  ```
  >> f3 = fspecial('disk',9);
  ```

- 'gaussian'. Crea un filtro paso bajo gaussiano. Como parámetros lleva el tamaño del filtro (un número si es cuadrado y 2 si es rectangular) y la desviación estándar. Por defecto estos valores son 3x3 y stddev=0.5.

  ```
  >> f4 = fspecial('gaussian',[7 7],0.8);
  ```

- 'laplacian'. Crea una máscara laplaciana (segunda derivada) de tamaño 3x3. Lleva un parámetro cuyo valores está en [0,1] para ajustar la forma de la máscara.
- 'log'. Calcula una máscara de tipo Laplaciana de gaussiana para calcular segundas derivadas. Necesita como parámetros el tamaño del filtro y la desviación de la función gaussiana.
- 'prewitt' y 'sobel'. Devuelven, respectivamente, máscaras 3x3 para el cálculo de los gradientes verticales de Prewitt y Sobel. Las máscaras para el gradiente horizontal son las mismas pero traspuestas.
- 'unsharp'. Produce un filtro que permite realzar los detalles de una imagen (le da más nitidez).
- 'motion'. Produce una máscara que simula un emborronamiento producido por el movimiento lineal de la cámara en una dirección dada.

4. Convolución

La convolución es una operación muy similar a la correlación. Se define como:

$$R(x,y)=\sum_{i=-N}^{N}\sum_{j=-N}^{N} I(x+i,y+j)*F(i,j)$$

De forma práctica, equivale a la correlación de la imagen con el filtro rotado 180 grados.

Para calcularla usamos también la función imfilter dándole como parámetro adicional 'conv':

```
>> r = imfilter(img,f,'conv');
```

Matlab dispone de dos funciones específicas para calcular la correlación y la convolución, similares a imfilter: conv2 y filter2. La diferencia es que imfilter está incluido en el toolbox de procesamiento de imágenes y estas dos funciones no necesitan dicho toolbox. Además, imfilter permite indicar opciones para el tratamiento de los bordes.

5. Núcleos separables

En algunos casos, es posible descomponer un filtro 2D como una composición de filtros 1D. Por ejemplo:

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} * \begin{bmatrix} 1/3 & 1/3 & 1/3 \end{bmatrix}$$

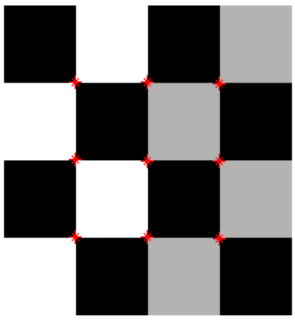
Es decir, el filtro 2D que calcula la media del vecindario 3x3 de un píxel se puede sustituir por un filtrado sucesivo con los otros dos filtros 1D, que es más eficiente. Siempre que sea posible será preferible usar filtros 1D que filtros 2D.

6. Puntos Esquina

Las esquinas pueden considerarse puntos de interés en un imagen. Por lo tanto una aguda detección de este tipo de punto de interés puede ser relevante para el procesamiento de la imagen y futuras decisiones. Para obtener las esquinas de una imagen Matlab tiene implementada la función 'corner':

```
>>I = checkerboard(50,2,2);
>>C = corner(I);
```

```
>>imshow(I);
>>hold on
>>plot(C(:,1), C(:,2), 'r*');
```



7. Ejercicios

1. Compara el resultado que producen el filtrado gaussiano y el filtro de mediana sobre las imágenes disney_r1.png ... disney_r5.png.
2. ¿Cómo se puede mejorar la calidad de las imágenes distorsion2.jpg, rostro1.png y rostro2.png?
3. Utiliza la correlación para buscar formas en una imagen. Para este ejercicio puedes usar las siguientes imágenes:
   - formas.png, estrella.png, ovalo.png, cuadrado.png, cuadrado2.png, cuadrado3.png
   - texto.png, letra_i.png, letra_k.png, letra_m.png, letra_o.png, letra_p.png
4. Analizar la imagen distorsion1.jpg y aplicar diferentes técnicas para mejorarla (eliminación del ruido). En concreto, prueba con suavizados gaussianos y con el filtro 'motion' de Matlab.

   ¿Se te ocurre alguna otra técnica que mejore sensiblemente la calidad de la imagen respecto de los filtrados propuestos?
5. Obtener sobre la imagen formas.png las esquinas usando el método de Harris.