

3 Procesamiento digital de imágenes

Una vez preparada la escena para ser capturada por las cámaras de Visión Artificial, se deberá de tratar el aspecto tanto de cuantificación como de codificación que sufre la señal de vídeo al ser introducida en el computador. Depositada la información, como una matriz o conjunto de matrices de valores discretos se procederá, tal cual se vio en las etapas de visión del capítulo 1, a un procesado de la imagen. Esta etapa, de bajo nivel, trata de obtener una nueva imagen que o bien mejore su calidad o bien destaque algún atributo primario de los objetos capturados. En el primer aspecto tratará de corregir las posibles faltas de iluminación uniforme, la eliminación del ruido o de aumentar el contraste en la imagen. Mientras en el segundo, su objetivo es realzar los bordes de los objetos, regularizar sus colores, destacar sus texturas, etc.

Las técnicas de procesamiento de las imágenes pueden ser clasificadas en dos grandes ramas: a) las procedentes del procesamiento de señales y b) aquellas que son de carácter heurístico y que nacieron para mejorar algún aspecto primario de la imagen.

En este capítulo se abordará el proceso de muestreo y cuantificación de la imagen y luego se examinarán algunas técnicas de procesado de imágenes procedentes del procesamiento de señales.

3.1 Muestreo y cuantificación

Las imágenes digitales son obtenidas por el proceso de muestreo y cuantificación de las señales analógicas adquirida a través de los sensores de visión. El muestreo consiste en la medición a intervalos de tiempo de la señal de vídeo analógica. La imagen es convertida en una matriz discreta de $M \times N$ píxeles. El muestreo está relacionado con la resolución espacial de la imagen. Ésta se define como el espacio muestreado sobre un objeto, a una distancia l respecto a la cámara, entre la distancia entre dos píxeles adyacentes.

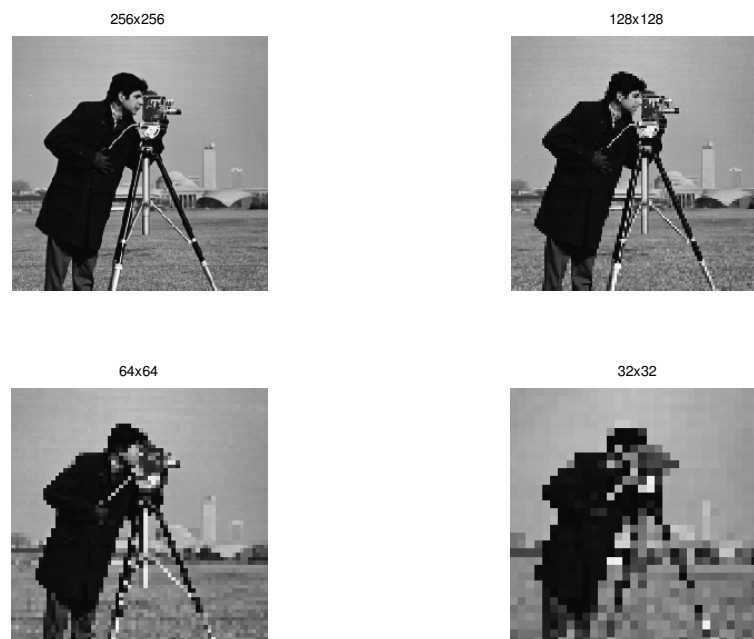


Figura 3. 1 Resoluciones distintas tomadas sobre un mismo escenario

Para determinar la resolución requerida, hay que determinar dos factores: 1) cuál es la característica más pequeña a observar y 2) cuál va a ser el campo de visión deseado a adquirir. Por tanto, habrá que determinar el factor de magnificación y el número de píxeles que debe de tener la cámara. Como regla general, se puede emplear que la característica más pequeña sea representada por, al menos, un cuadrado de 2×2 píxeles.

Por otro lado, el campo de visión suele venir dado por la capacidad del sistema de iluminación que por lo general, se pretenderá conseguir una intensidad luminosa uniforme en toda la superficie a inspeccionar por la cámara. Por tanto, se plantea el problema en los siguientes términos: defecto más pequeño a visualizar y el campo de visión a capturar por la cámara.

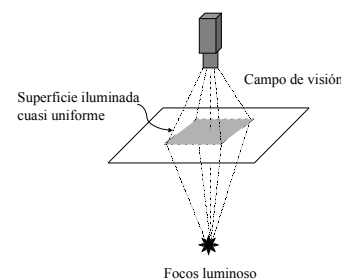
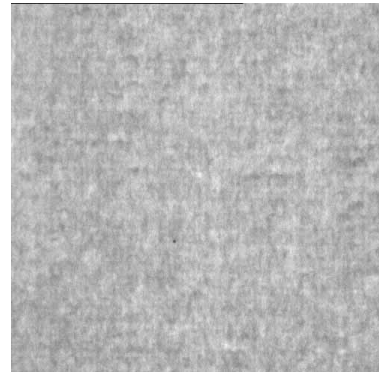


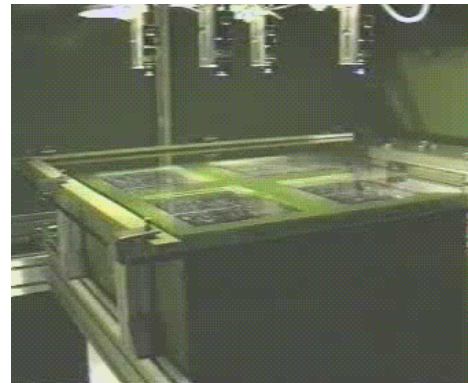
Figura 3. 2 Escenario de captura de la escena

Ejemplo 3.1

Para la inspección de pasta de papel se ha conseguido, según un modelo de simulación, una iluminación uniforme a contraluz de 200 mm x 275 mm (en una relación próxima a los $\frac{3}{4}$) y el defecto más pequeño a detectar tiene un área de 1 mm^2 . Con el objeto de reducir las aberraciones ópticas, se ha cerrado el diafragma con un elevado número F. Con ello y tras el análisis radiométrico se ha demostrado que si se emplea una cámara WATEC 902 con una lente de 16 mm, la pasta de papel debe de estar alrededor de los 700 mm en vertical. Determinar si es correcta la elección realizada.



Según el fabricante de la cámara, ésta tiene para el estándar CCIR 582 filas por 752 columnas y el tamaño del píxel es $8.3\mu\text{m}$ por $8.6\mu\text{m}$. Tomando como propósito un entorno de 3×3 píxeles para el defecto de 1 mm^2 , el factor de magnificación ponderado será:



$$M^2 = \frac{9 \cdot 8.3 \cdot 10^{-6} \cdot 8.6 \cdot 10^{-6}}{10^{-6}} \quad M = 0.0253$$

La superficie a inspeccionar será:

$$X = \frac{582 \cdot 8.3 \cdot 10^{-3}}{M} = 190.6 \text{ mm} \quad Y = \frac{752 \cdot 8.6 \cdot 10^{-3}}{M} = 255.2 \text{ mm}$$

La cámara deberá estar situada a:

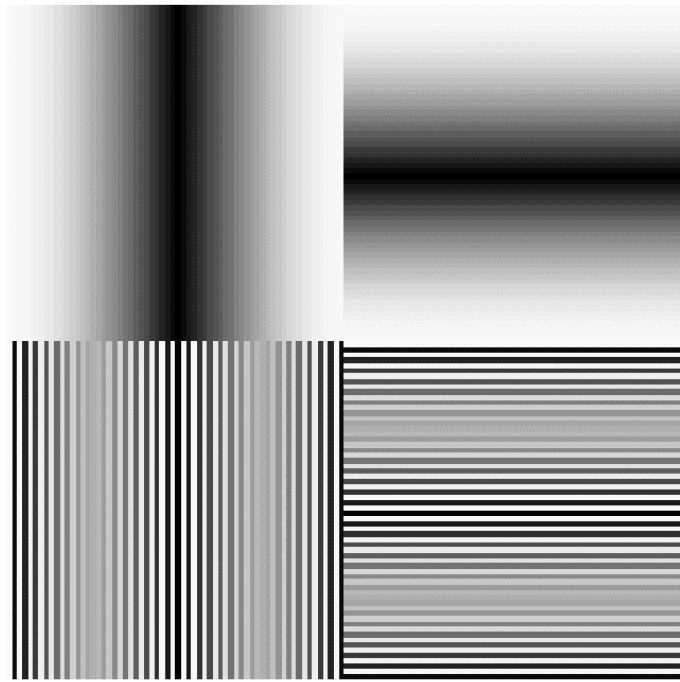
$$Z = \frac{f}{M} = 632 \text{ mm}$$

En el caso de obtener una resolución elevada, se puede optar por dos soluciones:

1. Si es posible parar la escena, puede resultar ser más ventajoso utilizar varias cámaras de baja resolución en lugar de una de alta resolución, pues se consigue disminuir anchos de banda puntuales y resulta ser una solución más económica.
- 2) Si la escena no se puede detener y el número de detectores a utilizar es elevado, bien porque la arquitectura de iluminación sólo permite iluminar uniformemente una estrecha banda, o bien porque se quiere cubrir el ancho de la banda del material a inspeccionar, las cámaras lineales y TDI son las más apropiadas. Estas cámaras tienen la ventaja de poner decenas de miles de píxeles en línea, cubriendo toda la cubierta de la superficie y con una resolución muy alta. La adquisición de una sola línea permite un ancho de banda razonable, aun con decenas de miles de detectores. Por otro lado, las altas velocidades hacen que los tiempos de exposición sean muy pequeños, obligando a aumentar la

intensidad luminosa. Esta consecuencia, puede ser disminuida con cámaras TDI, siendo éstas las más utilizadas en inspección de superficies continuas en línea.

Desde el punto de vista frecuencia, el teorema de Shanon afirma que la frecuencia de muestreo debe ser al menos el doble que al ancho de banda de la señal a muestrear. En una imagen capturada, sin cambios abruptos de intensidad, las componentes de alta frecuencia de su transformada de Fourier tienden a ser nulos. Por el contrario, si la imagen contiene muchos pequeños detalles y una gran cantidad de bordes de los objetos proyectados, el espectro de alta frecuencia es de elevado nivel. Considerando que la imagen tiene un espectro de ancho de banda limitado, W , la frecuencia de muestreo debe de ser al menos del doble, $2W$. En la cuantificación de la imagen, la unidad de frecuencia especial es un ciclo por píxel. Por tanto, desde el punto de vista del teorema, la mayor frecuencia espacial representada es la de $\frac{1}{2}$ ciclo por píxel. El armónico de mayor frecuencia requiere de dos píxeles; realizando una transición de blanco-negro.



Por esta razón, se decía anteriormente que el detalle más pequeño debería de tener

un entorno de 2×2 . En la figura se ha representado el armónico fundamental junto con la componente de mayor frecuencia, tanto en la frecuencia horizontal como en vertical.

Figura 3.3. Componentes de Fourier a) Armónico fundamental, b) Componente de alta frecuencia

La segunda operación es la cuantificación de la señal, que consiste en el número finito de valores de cada píxel. Los niveles de cuantificación suelen ser potencias de 2 para facilitar su almacenamiento en el computador. El número de niveles posibles define la resolución radiométrica. Cuando las imágenes solo tienen información sobre la luminancia se habla de una imagen en niveles de grises. En este caso, la cuantificación del píxel es de tipo escalar. Si el escalar sólo tiene dos valores, se dice que la imagen es binaria. El valor del cero es negro y el uno es blanco.

En general, se suele emplear un byte de cuantificación por píxel, consiguiendo 256 niveles de grises. El cero será el color negro y el valor 255 corresponderá con el blanco. Entre el cero y el valor 255 estarán los diversos valores de grises. Para el caso del color, la cuantificación se vuelve vectorial; por cada píxel se representara una terna de valores que haga reflejar la luminancia y crominancia en cada píxel. La terna dependerá del sistema de representación del color: RGB, L^*a^*b , HSI,... Comúnmente, se empleará el sistema RGB con 1 byte por cada color, de forma que se obtendrá 16

millones de colores. Las aplicaciones industriales suelen emplear una resolución radiométrica de 256 niveles de grises o de 16 millones de colores. Evidentemente, si la aplicación exige más resolución, el volumen de información se incrementa exponencialmente.

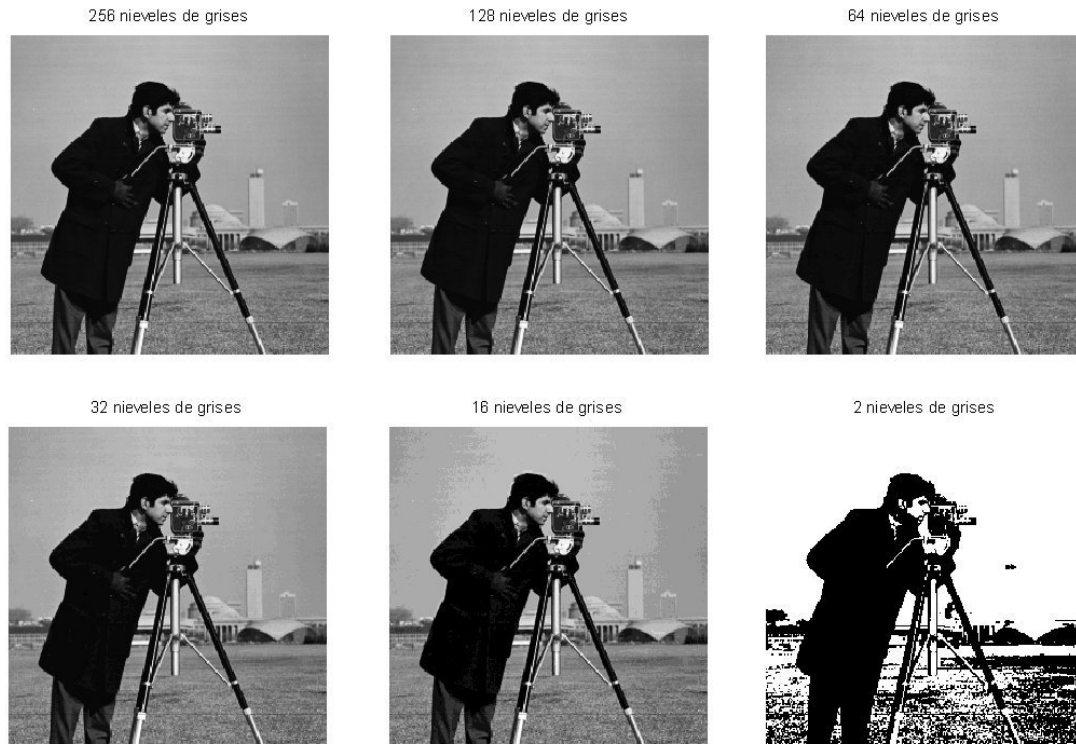


Figura 3. 4 Niveles de cuantificación

3.2 Distancias entre píxeles

Antes de entrar en el procesado, se definen ciertas relaciones que se establecen entre los píxeles de una imagen y que serán aplicadas en las técnicas de procesamiento de las imágenes. En primer lugar, se mencionarán los diferentes conceptos de distancias, para una vez establecidas pasar a las relaciones de vecindad.

3.2.1 Relaciones de distancias

La distancia más usual entre dos píxeles es la distancia geométrica o distancia euclídea. La distancia euclídea entre el píxel p de coordenadas (x,y) y el píxel q de posición (s,t) será:

$$D_e(p, q) = \sqrt{(x-s)^2 + (y-t)^2} \quad (3.1)$$

El problema de esta métrica está en el tiempo de cálculo que requiere. Por dicha razón también se suele emplearse, en imágenes digitales, la distancia rectangular o *Manhattan*, definida como:

$$D_m(p, q) = (|x - s| + |y - t|) \tag{3.2}$$

Y la distancia de Tchebychev, tomada como:

$$D_t(p, q) = \max(|x - s|, |y - t|) \tag{3.3}$$

Para realizar una comparativa entre estas tres distancias, véanse en la figura adjunta el lugar geométrico de estas distancias en el entorno de vecindad de un píxel centrado en la cuadrícula:

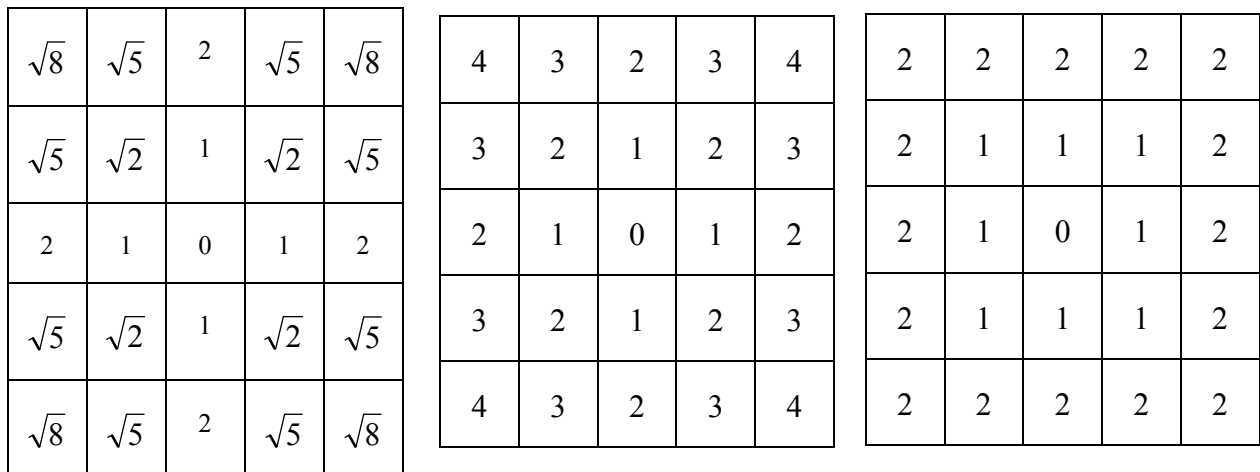


Figura 3. 5. a)Euclídea b)Rectangular c)Tchebychev

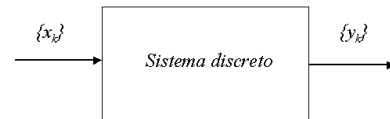
3.2.2 Relaciones de conectividad

Mediante el concepto de conectividad se quiere expresar que dos píxeles pertenecen al mismo objeto, por lo que está relacionado con la vecindad. Se define el conjunto de los 4-vecinos de un píxel dado p , definido como $N_4(p)$, como el conjunto formado por los píxeles de arriba, de abajo, de izquierda y a la derecha. Desde un punto de vista más formal, serán aquellos píxeles que respecto de p tengan una distancia rectangular igual a la unidad, $D_m(p, q) = 1$. La vecindad a 8, denotado por $N_8(p)$, está formado por el conjunto de píxeles que estén a una distancia de Tchebychev igual a la unidad, $D_t(p, q) = 1$, esto es, el entorno total de vecindad del píxel.

3.3 Procesamiento de las imágenes

La Teoría de Señales y Sistemas da un marco de trabajo para el procesamiento digital de las imágenes. Con este objetivo, se procederá a extender los conocimientos dados sobre señales unidimensionales y sistemas discretos monovariantes, estudiados en anteriores asignaturas, a las imágenes digitales. Las imágenes digitales son consideradas como señales discretas bidimensionales, las cuales son procesadas por sistemas discretos o filtros.

Según la Teoría de la Señal, una señal discreta unidimensional que es procesada por un filtro discreto lineal (*Linear Time Invariant*, LTI), la secuencia de salida es explicada por el proceso de convolución discreta entre la entrada y la secuencia de ponderación del filtro:



$$\{y_k\} = \{g_k\} * \{x_k\} = \{x_k\} * \{g_k\} \quad (3.4)$$

$$y_k = \sum_{n=-\infty}^{\infty} g_{k-n} x_n = \sum_{n=-\infty}^{\infty} x_{k-n} g_n \quad (3.5)$$

Se denomina secuencia de ponderación a la que se obtiene en la salida del sistema cuando se aplica a la entrada una secuencia tipo impulso.

$$\{g_k\} = \{\dots, g_{-2}, g_{-1}, g_0, g_1, g_2, \dots\} \quad (3.6)$$

Si el sistema es causal, g_k es igual a cero para $k < 0$.

Ejemplo 3.2

Dada la ecuación en diferencia:

$$y_k = \frac{1}{4} x_k + \frac{1}{2} x_{k-1} + \frac{1}{4} x_{k-2}$$

obtener la secuencia de ponderación y determinar la salida ante una entrada en escalón.

Para obtener la secuencia de ponderación habrá de aplicarle una entrada impulsional, de forma que la entrada será del tipo $\{x_k\} = \{1, 0, 0, 0, \dots\}$. La evolución de la salida puede ser realizada a través de una tabla, en las que se van obteniendo los resultados temporales al aplicar la ecuación en diferencias:

K	x_k	x_{k-1}	x_{k-2}	y_k
0	1	0	0	0.25
1	0	1	0	0.5
2	0	0	1	0.25

Luego la secuencia de ponderación del sistema LTI es:

$$\{g_k\} = \{1 \ 2 \ 1\} \frac{1}{4}$$

Para calcular la respuesta en escalón se aplicará el operador convolución discreta, donde la señal de entrada será $\{x_k\} = \{1,1,1,1,\dots\}$:

$$y_0 = \sum_{n=0}^2 x_{k-n} g_n = 0.25$$

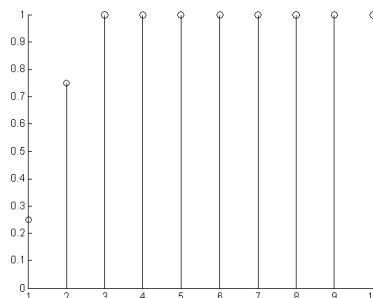
$$y_1 = \sum_{n=0}^7 x_{k-n} g_n = 0.25 + 0.5 = 0.75$$

$$y_2 = \sum_{n=0}^7 x_{k-n} g_n = 0.25 + 0.5 + 0.25 = 1$$

...

Resolución en MATLAB

```
>> g = [1;2;1]/4;
>> x = ones(10,1);
>> y = conv(x,g);
>> stem(y(1:10));
```



Cuando la secuencia de ponderación de un filtro es finita se le denomina filtro de tipo FIR (*Finite Impulse Response*). Si la secuencia de ponderación tiene infinito términos, el filtro se denomina IIR (*Infinite Impulse Response*).

De otro lado, la suma de los coeficientes de la máscara del filtro debe ser unitaria, para preservar el rango dinámico de la secuencia de entrada:

$$\sum_{k=0}^n g_k = 1 \quad (3.7)$$

n será el grado del filtro FIR.

Estos razonamientos se extienden a señales bidimensionales discretas, aplicándose, por tanto, a las imágenes digitales. En este caso, la señal de entrada será la imagen discreta a procesar que es convolucionada con el filtro lineal discreto y la señal de salida será la imagen resultante. La expresión de la convolución discretas 2D será:

$$y_{k,l} = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x_{k-m,l-n} g_{m,n} \quad (3.8)$$

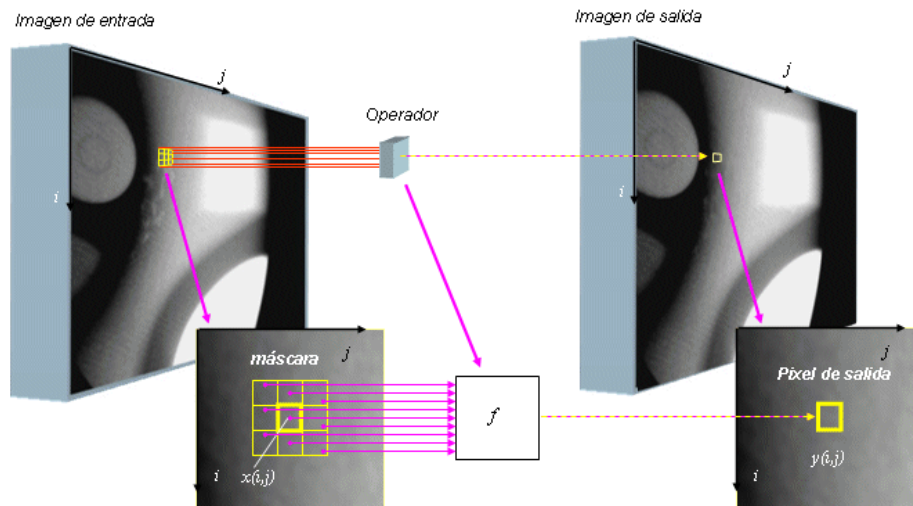
Normalmente, el filtro discreto está constituido por una secuencia de ponderación finita FIR, del tipo 3x3 o de 5x5. A los coeficientes de estos filtros se les llama máscaras de convolución. El proceso de convolución se presenta como una combinación lineal de los píxeles del entorno ponderados por los elementos de la secuencia de ponderación o máscara de convolución, de forma parecida al caso unidimensional. Por ejemplo, para el caso de un filtro de 3x3, el valor de píxel de salida (k,j) será:

$$\begin{aligned} y_{k,l} = \sum_{m=-1}^1 \sum_{n=-1}^1 x_{k-m,l-n} g_{m,n} = & x_{k,l} \cdot g_{0,0} + x_{k,l-1} \cdot g_{0,1} + x_{k,l+1} \cdot g_{0,-1} + \\ & x_{k+1,l} \cdot g_{-1,0} + x_{k+1,l-1} \cdot g_{-1,1} + x_{k+1,l+1} \cdot g_{-1,-1} + \\ & x_{k-1,l} \cdot g_{1,0} + x_{k-1,l-1} \cdot g_{1,1} + x_{k-1,l+1} \cdot g_{1,-1} \end{aligned} \quad (3.9)$$

Las gráficas del filtro y la imagen en subíndices y expresadas en forma matricial corresponderán a:

$$\begin{pmatrix} g_{-1,-1} & g_{-1,0} & g_{-1,1} \\ g_{0,-1} & g_{0,0} & g_{0,1} \\ g_{1,-1} & g_{1,0} & g_{1,1} \end{pmatrix} \quad \begin{pmatrix} x_{k-1,l-1} & x_{k-1,l} & x_{k-1,l+1} \\ x_{k,l-1} & x_{k,l} & x_{k,l+1} \\ x_{k+1,l-1} & x_{k+1,l} & x_{k+1,l+1} \end{pmatrix}$$

La convolución discreta 2D se presenta como una combinación lineal, donde el valor del píxel de salida (k,l) está definido por la multiplicación de sus vecinos ponderados por los coeficientes de la máscara de convolución. El valor del píxel de entrada del vecino a (k,l) es multiplicado por el opuesto de la máscara de convolución. Por ejemplo, el vecino $(k+1,l+1)$ está ponderado por el coeficiente $(-1,-1)$ de la máscara.



Ejemplo 3.3

Determinar la imagen de salida cuando ésta es procesada por un filtro binomial de 3 x 3.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 200 & 200 & 200 & 0 \\ 0 & 200 & 0 & 200 & 0 \\ 0 & 200 & 200 & 200 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \frac{1}{16} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12.5 & 37.5 & 50 & 37.5 & 12.5 & 0 \\ 0 & 37.5 & 100 & 125 & 100 & 37.5 & 0 \\ 0 & 50 & 125 & 150 & 125 & 50 & 0 \\ 0 & 37.5 & 100 & 125 & 100 & 37.5 & 0 \\ 0 & 12.5 & 37.5 & 50 & 37.5 & 12.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



Resolución en MATLAB

```

>> g= [1;2;1]/4;
>> mascara = conv2(g,g');
>> imagen = zeros(5,5);imagen(2:4,2:4)=200;imagen(3,3)=0;
>>imgSal = conv2(imagen,mascara)
    
```

Del ejemplo se observa que la aplicación del operador convolución sobre la imagen produce una imagen resultado de mayor dimensión que la entrada. La imagen salida será la suma de filas de la entrada más la del filtro menos la unidad. Igualmente sucede con las columnas. Si $K \times L$ es el rango de la matriz de entrada y $M \times N$ la máscara, el rango de la salida es $(K+M-1) \times (L+N-1)$. Este efecto se debe a cuando el operador

convolución pasa por los bordes de la imagen. En los píxeles de los bordes cuando se pivota la máscara, los vecinos del píxel que no existen, por defecto, son considerados nulos. En los algoritmos que implementan esta operación el efecto de los bordes son tratados de tres maneras distintas:

1. Se considera que los píxeles vecinos no existentes son cero, un valor dado por el usuario o se toma un valor intermedio de los que existen y la matriz de salida es $(K+M-1) \times (L+N-1)$.
2. El rango de la matriz de salida es igual al de la entrada, $(K \times L)$. Se hace la misma convolución que en el apartado anterior pero eliminando las filas y columnas exteriores.
3. La convolución se hace sólo en la parte central de la imagen de entrada, esto es, se hace una submatriz tal que al convolucionar todos los píxeles de éstas tienen vecinos conocidos. El rango de la matriz de salida es $(K-M+1) \times (L-N+1)$.

Ejemplo 3.4

Determinar el resultado de la convolución discreta 2D para el filtro FIR binomial $\{1,2,1\}$ y su transpuesta.

$$(1 \ 2 \ 1) * \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Resolución en MATLAB

```
>> g= [1;2;1]/4;
>> conv2(g,g')
```

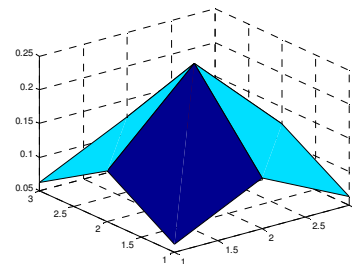
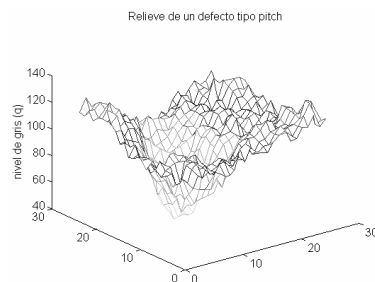
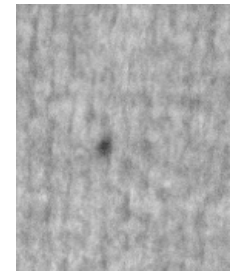
Una ayuda a la comprensión del efecto de la convolución es dado por el operador correlación discreta 2D. Esta transformación se define como:

$$y_{k,l} = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x_{k+m,l+n} g_{m,n} \quad (3.10)$$

Nótese que para máscaras de convolución con simetría par, la convolución y la correlación son operaciones idénticas. Para este caso, la convolución es aplicada cómo búsqueda de patrones en la imagen. Si la forma de la máscara y de la imagen son parecidas la imagen de salida será reforzada en estos píxeles, ya que existe correlación. En caso contrario, los píxeles de salida que tienen una forma contraria serán restados.

Ejemplo 3.5

Para determinar impurezas en la pasta de papel se ha empleado un filtro binomial. Su fundamento se encuentra en la correlación existente entre los tipos de defectos y la forma de la máscara. Mientras, los defectos en el papel tienen la forma de hoyo, el filtro binomial se aparece a un hongo. Por tanto su correlación es negativa, acentuando el nivel de gris a la baja cuando hay un defecto en el papel.



El operador correlación es empleado en las técnicas de Visión Artificial como instrumento para la búsqueda de patrones en la imagen (*Pattern Matching*). Por ejemplo, localizar fechas de caducidad, formas geométricas de objetos, etc. No obstante, la variabilidad de la iluminación en la escena requiere que la correlación sea independiente del nivel de gris y que dependa sólo de la forma. Para solucionarlo se emplean operadores de correlación normalizada¹. Los fabricantes de tarjetas de procesamiento de imágenes implementan la operación correlación en vez de la convolución, ahorrando tiempo computacional en el filtrado lineal. Esta consecuencia es por que las mayorías de las máscaras de convolución tienen simetría par. En caso de simetría impar, el resultado de la convolución y de la correlación son iguales pero de signo contrario.



Figura 3. 6. Aplicaciones de *pattern matching*

No obstante, la interpretación más fructífera de la convolución se da en la respuesta en frecuencia de los filtros.

¹ Véase la bibliografía especializada.

3.4 Respuesta en frecuencia y transformadas de Fourier

Para señales unidimensionales y considerando normalizado el periodo de muestreo a 1 segundo, ya se había definido la respuesta en frecuencia de un filtro lineal discreto como:

$$G(\omega) = \sum_{n=-\infty}^{\infty} g_n e^{-j\omega n} \quad (3.11)$$

Ejemplo 3.6

Determinar la respuesta en frecuencia de un sistema discreto cuya secuencia de ponderación es $\{1/4, 1/2, 1/4\}$.

Al aplicar (3.11) sobre la secuencia de ponderación dada:

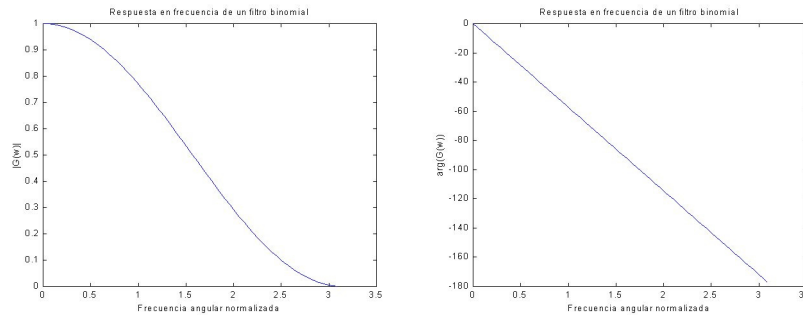
$$G(\omega) = \sum_0^2 g_n e^{-j\omega n} = \frac{1}{4} + \frac{1}{2} e^{-j\omega} + \frac{1}{4} e^{-j2\omega}$$

dando valores de 0 hasta la frecuencia de Nyquist, π/T , se conseguirá la respuesta. Aunque $G(\omega)$ es una función continua, con el propósito de representarla en el diagrama de Bode se hacen, por ejemplo, 64 intervalos entre 0 y π , confeccionándose la tabla siguiente:

ω	$G(\omega)$	$ G(\omega) $	$\arg(G(\omega))$
0	1	1	0
$\pi/64$	0.99-j0.049	0.999	-2.81°
$2\pi/64$	0.99-j0.097	0.997	-5.62°
...
$63\pi/64$	-0.0006-j0.00003	0.0006	-178°

Resolución en MATLAB

```
>> [G,W] = freqz ([1/4,1/2,1/4], 1,128);
>> plot(W,abs(G));
>> plot(W,angle(G).*(180/pi));
```



En este sentido, aunque la respuesta en frecuencia es continua, los algoritmos de cálculo son discretos, por tanto, si la frecuencia va desde cero hasta la frecuencia de Nyquist y se hacen $K/2$ de particiones para este rango de frecuencias, este operador podrá redefinirse como:

$$G_k = \sum_{n=0}^N g_n e^{-j\frac{2\pi \cdot k}{K}n} \quad k = 0,1,2,\dots, K-1 \quad (3.12)$$

Recuérdese el carácter simétrico de la respuesta en frecuencia entre el rango de frecuencias entre 0 y la frecuencia de Nyquist y de esta frecuencia a la de muestreo. La antitransformada quedará aproximada por la algoritmia numérica:

$$g_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} G(\omega) e^{j\omega \cdot n} d\omega \cong \frac{1}{K} \sum_{k=0}^{K-1} G_k e^{j\frac{2\pi \cdot k}{K}n} \quad n = 0,1,2,\dots, N-1 \quad (3.13)$$

La forma de asociar la secuencia de ponderación $\{g_n\}$ a la respuesta en frecuencia G_k puede ser extendida a cualquier secuencia que cumpla determinadas propiedades. Así se construye lo que se denomina la transformada discreta de Fourier de una secuencia, a ésta se la conoce por las siglas DFT (*Discrete Fourier Transform*). De este modo, dada una secuencia, $\{x_n\}$, se define su transformada de Fourier como la función:

$$X_k = \sum_{n=0}^N x_n e^{-j\frac{2\pi \cdot k}{K}n} \quad k = 0,1,2,\dots, K-1 \quad (3.14)$$

Ésta es igualmente periódica, de ciclo 2π , y en las condiciones antes indicadas es un desarrollo en serie de Fourier, lo que permite definir la transformada inversa:

$$x_n = \frac{1}{K} \sum_{k=0}^{K-1} X_k e^{j \frac{2\pi \cdot k}{K} \cdot n} \quad n = 0, 1, 2, \dots, N-1 \quad (3.15)$$

La transformada de Fourier de una secuencia es una función compleja, siendo usualmente representada gráficamente por sus curvas de módulo y de argumento. Al igual que ya se vio en la Teoría de Control, la utilidad más importante que se va a hacer de la transformación de Fourier es la que se deriva de la relación existente entre las transformadas de Fourier de las secuencias de entrada y de salida de un filtro LTI y su respuesta en frecuencia, de manera que:

$$Y(\omega) = G(\omega)X(\omega) \quad (3.16)$$

Esta relación permite tratar a los filtros discretos de una manera similar con lo que se procedía con los continuos; dando un procedimiento para determinar la secuencia de salida a partir de la respuesta en frecuencia, ver figura 3.7. Conocida la respuesta en frecuencia del filtro y la transformada discreta de la secuencia de entrada, se determinará la transformada de la secuencia de salida y posteriormente se aplicará la transformada inversa, obteniendo la secuencia de la salida.

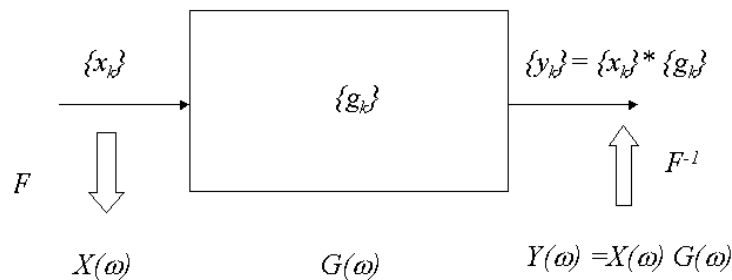


Figura 3.7 Relación fundamental de los sistemas discretos

Ejemplo 3.7

Dado la respuesta en frecuencia de un filtro discreto unidimensional de orden 2 (ver ejemplo 3.6):

$$G_k = \frac{1}{4} + \frac{1}{2} e^{-j \frac{2\pi}{K} k} + \frac{1}{4} e^{-j \frac{4\pi}{K} k} \quad k = 0, 1, 2, \dots, K-1$$

y siendo $\{x_k\} = \{1, 1, 0, 0, \dots\}$ la excitación al sistema, calcular la secuencia de salida.

En primer lugar se procederá a calcular la DFT de la secuencia de entrada:

$$X_k = 1 + e^{-j\frac{2\pi}{K}k} \quad k = 0, 1, 2, \dots, K-1$$

Aplicando la relación fundamental visto en la ec.(3. 16) se tendrá que:

$$Y_k = \left(1 + e^{-j\frac{2\pi}{K}k}\right) \left(\frac{1}{4} + \frac{1}{2}e^{-j\frac{2\pi}{K}k} + \frac{1}{4}e^{-j\frac{4\pi}{K}k}\right) = \frac{1}{4} + \frac{3}{4}e^{-j\frac{2\pi}{K}k} + \frac{3}{4}e^{-j\frac{4\pi}{K}k} + \frac{1}{4}e^{-j\frac{6\pi}{K}k}$$

Por tanto, la secuencia de salida valdrá:

$$\{y_k\} = \left\{\frac{1}{4}, \frac{3}{4}, \frac{3}{4}, \frac{1}{4}\right\}$$

Resolución en MATLAB

```
>> Xw = fft( [1;1], 64);
>> Gw = fft( [1;2;1]/4, 64);
>> Yw = Xw .* Gw;
>> yk = ifft( Yw )
    0.25
    0.75
    0.75
    0.25
    0
    ...
```

3.4.1 Extensión a imágenes digitales

La respuesta en frecuencia de un filtro LTI 2D de respuesta finita, FIR, aplicado sobre una imagen es la extensión de la definición del operador respuesta en frecuencia sobre la máscara de convolución. Dada una máscara de convolución, $g_{n,m}$, su respuesta en frecuencia es:

$$G_{k,l} = \sum_{m=0}^M \sum_{n=0}^N g_{m,n} e^{-j\frac{2\pi \cdot m \cdot k}{M}} e^{-j\frac{2\pi \cdot n \cdot l}{N}} \quad k = 0, 1, 2, \dots, K-1 \quad l = 0, 1, 2, \dots, L-1 \quad (3. 17)$$

Ejemplo 3.7

Determinar la respuesta frecuencia del filtro binomial 2D:

$$\{g_{m,n}\} = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

Al aplicar la definición sobre la máscara de convolución dada:

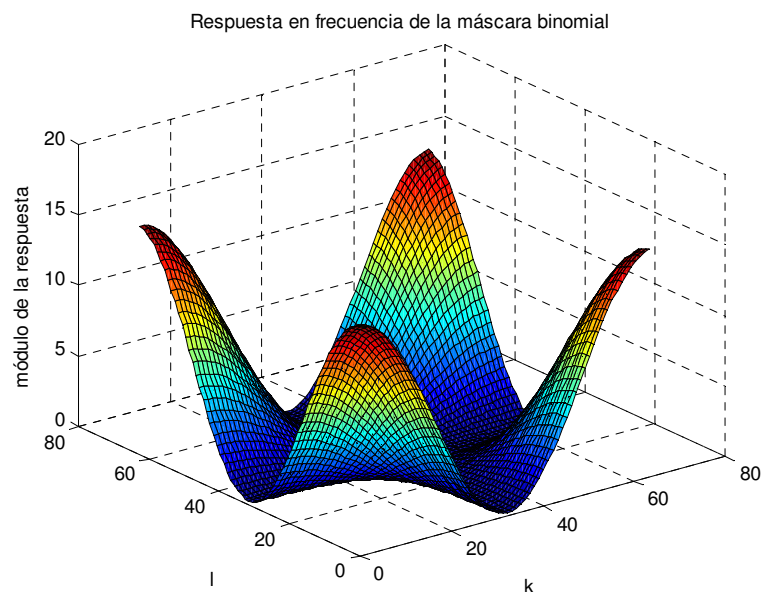
$$G_{k,l} = \sum_0^2 \sum_0^2 g_{m,n} e^{-j\frac{2\pi \cdot m}{K} \cdot k} e^{-j\frac{2\pi \cdot n}{L} \cdot l} = 1 + 2e^{-j\frac{2\pi}{L}l} + e^{-j\frac{4\pi}{L}l} +$$

$$2e^{-j\frac{2\pi}{K}k} + 4e^{-j\frac{2\pi}{K}k} e^{-j\frac{2\pi}{L}l} + 2e^{-j\frac{2\pi}{K}k} e^{-j\frac{4\pi}{L}l} +$$

$$e^{-j\frac{4\pi}{K}k} + 2e^{-j\frac{4\pi}{K}k} e^{-j\frac{2\pi}{L}l} + e^{-j\frac{4\pi}{K}k} e^{-j\frac{4\pi}{L}l}$$

Resolución en MATLAB

```
>> V=fft2(conv2([1 2 1],[1 2 1]'),64,64);
>> surf(abs(V))
```



El nivel de resolución se vuelve tan enredoso que sólo se puede realizar desde la simulación. En la presentación con Matlab obsérvese que se ha empleado para K y L un valor en potencia de 2, como es 64. Los valores de K y L son elegidos de esta manera para optimizar el tiempo de cálculo de la transformada de Fourier, llamado algoritmo rápido de esta transformación. En la representación sólo se ha presentado el módulo. Al igual que en el caso unidimensional, existe simetría tanto en la dirección k como en l .

Las imágenes discretas también poseen su transformada de Fourier. Cualquier imagen digital puede ser representada por una serie de exponenciales discretas:

$$X_{k,l} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{m,n} e^{-j\frac{2\pi \cdot m \cdot k}{K}} e^{-j\frac{2\pi \cdot n \cdot l}{L}} \quad k = 0,1,2,\dots,K-1 \quad l = 0,1,2,\dots,L-1 \quad (3.18)$$

Dada la transformada de Fourier de la imagen también se puede recuperar ésta a través de su antitransformada:

$$x_{m,n} = \frac{1}{K \cdot L} \sum_{k=0}^{K-1} \sum_{l=0}^{L-1} X_{k,l} e^{j\frac{2\pi \cdot k \cdot m}{K}} e^{j\frac{2\pi \cdot l \cdot n}{L}} \quad m = 0,1,2,\dots,M-1 \quad n = 0,1,2,\dots,N-1 \quad (3.19)$$

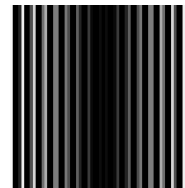
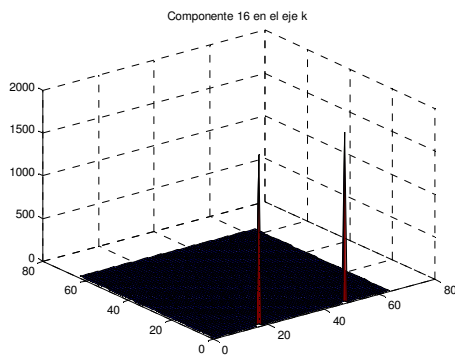
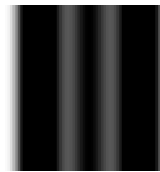
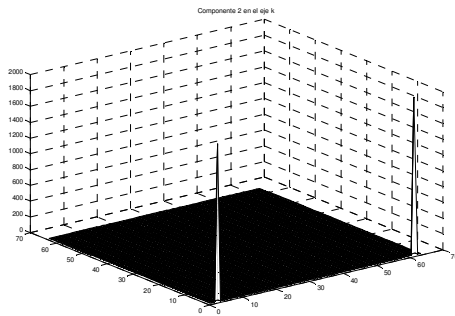
La transformada de Fourier obtiene el espectro frecuencial de la imagen. A medida que aumenta la frecuencia de las componentes del espectro, éstas indican más transiciones de luminancia en menos píxeles de la imagen. Imagínese que sólo se utilice una única componente del desarrollo de Fourier, por ejemplo, un armónico en la frecuencia de la dirección k . Al realizar su antitransformada, mostrará los cambios producidos por el armónico en la dirección k en un determinado conjunto de píxeles. Al aumentar la frecuencia de esta componente, las transiciones de grises serán mayores en el conjunto de píxeles seleccionados.

Resolución en MATLAB

```

>> V=zeros(64,64);V(1,1+2)=2000;V(1,64-2)=2000;
>> surf(V);title('Componente 2 en el eje k');
>> imshow(iff2(V,64,64));
>> V=zeros(64,64);V(1,1+16)=2000;V(1,64-16)=2000;
>> surf(V);title('Componente 16 en el eje k');
>> imshow(iff2(V,64,64));

```



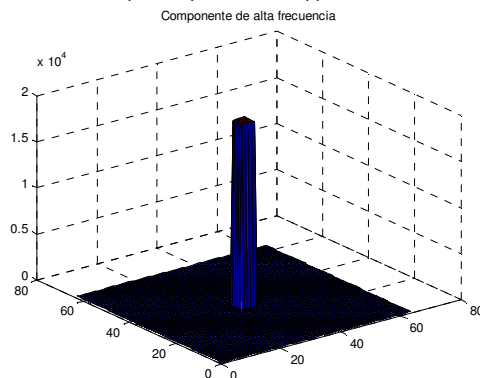
Las mayores transiciones se darán en las altas frecuencias definiendo la máxima resolución del sistema de formación de las imágenes.

Resolución en MATLAB

```

>> V=zeros(64,64);V(30:34,30:34)=20000;
>> surf(V);title('Componente de alta frecuencia');
>> imshow(iff2(V,64,64))

```



La transformada de Fourier muestra que una imagen puede ser construida por la combinación de armónicos de frecuencias verticales y horizontales.

3.4.2 Aplicaciones

De la interpretación de la transformada de Fourier de las imágenes se deducen aplicaciones inmediatas. Una primera aplicación sería la eliminación del ruido de la imagen. Es conocido que el espectro del ruido corresponde a la alta frecuencia, por tanto, si se elimina estas componentes se hará cancelado el ruido. La desventaja de este proceder es que eliminará también los detalles finos de la imagen.

Resolución en MATLAB

```
>> imagen=imnoise(imread('cameraman.tif'));imshow(imagen);  
>>V=fft2(double(imagen)); V(128-120:128+120,128-120:128+120)=0;  
>> imshow(uint8(iff2((V))))
```



Otra aplicación sería realzar los bordes de la imagen. Los contornos son transiciones en la luminancia de alta frecuencia, la eliminación de las componentes de baja frecuencia permitirá destacar los bordes.

Resolución en MATLAB

```
>> imagen= imread('cameraman.tif');imshow(imagen);
>>V=fft2(double(imagen));
>>V(1:10,1:10)=0;V(256-9:256,1:10)=0;
>>V(256-9:256,1:10)=0;V(256-9:256,256-9:256)=0;
>> imshow(uint8(iff2((V))))
```



Sin embargo, esta manera de actuación no se emplea por el alto coste computacional. Muchas de las técnicas de procesamiento emplean la relación fundamental de los sistemas discretos. Se diseñan máscaras de convolución cuya respuesta en frecuencia correspondan a filtros pasos bajos, pasos bandas o pasos altos, eligiendo el espectro de la imagen de entrada que se pretende seleccionar. Debido a la profundidad de estas técnicas se abordarán en el próximo capítulo.

3.5 Problemas

1. ¿Cómo determinar el proceso de cuantificación de una imagen?.¿Y el de codificación?.
2. El procesamiento lineal de las imágenes digitales.
3. Convolución y correlación 2D, conclusiones y aplicaciones.

4. Respuesta en frecuencia de la máscara de convolución:
$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

5. Relaciones entre el espectro de la imagen y el tipo de escena capturada.

Ejercicio 1

Dada la siguiente imagen, calcular para el píxel marcado el resultado de la convolución con las dos máscaras de Prewitt.

$$\begin{pmatrix} 1 & 1 & 10 & 10 & 10 \\ 1 & 1 & 10 \bullet & 10 & 10 \\ 1 & 1 & 10 & 10 & 10 \\ 1 & 1 & 10 & 10 & 10 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

$$\frac{\partial f}{\partial x} \qquad \frac{\partial f}{\partial y}$$

Derecho de Autor © 2007 Carlos Platero Dueñas.

Permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation; sin secciones invariantes, sin texto de la Cubierta Frontal, así como el texto de la Cubierta Posterior. Una copia de la licencia es incluida en la sección titulada "Licencia de Documentación Libre GNU".

La Licencia de documentación libre GNU (GNU Free Documentation License) es una licencia con [copyleft](#) para [contenidos abiertos](#). Todos los contenidos de estos apuntes están cubiertos por esta licencia. La versión 1.1 se encuentra en <http://www.gnu.org/copyleft/fdl.html>. La traducción (no oficial) al castellano de la versión 1.1 se encuentra en <http://www.es.gnu.org/Licencias/fdles.html>