

Una biblioteca para procesamiento de imagen: scimagen

Julio Cornejo Herrera, Adriana Lara López, Ricardo Landa Becerra y
Luis Gerardo de la Fraga *

Sección de Computación. Departamento de Ingeniería Eléctrica
CINVESTAV-IPN
Av. Instituto Politécnico Nacional 2508. 07300 México, D.F.

Resumen

En este trabajo se describe una nueva biblioteca para procesamiento de imagen. Las motivaciones para la creación de esta nueva biblioteca han sido: contar con el código fuente de ella y ponerlo disponible libremente a la comunidad científica, tener una herramienta propia para la enseñanza de procesamiento de imagen y de programación en el lenguaje C, y tener una plataforma propia para la investigación en técnicas de procesamiento de imagen. Aquí se desarrollarán los tópicos para la docencia, se presentan resultados de la investigación en la comparación de distintos perfiles usados para el filtraje de imágenes en el dominio de la frecuencia. Finalmente se presentan la realización de dos algoritmos para análisis de imagen: la extracción de formas y la representación de una forma a través de su esqueleto.

Palabras clave: procesamiento de imagen, análisis de imagen, restauración, filtraje en frecuencia, visualización, transformada de Hartley, transformada coseno discreta.

1. Introducción

Actualmente existen varios paquetes de uso libre para el procesamiento digital de imágenes (PDI) en Internet, ejemplos de ellos son Xite [1] de la universidad de Oslo, y Khoros [2]. Khoros es un ambiente de desarrollo gráfico, que incluye un lenguaje de programación visual, generadores de código para extender el lenguaje visual y adicionar nuevos paquetes de aplicación al sistema, un paquete interactivo de despliegue de imágenes, una biblioteca extensiva de rutinas para

procesamiento de señales y de imágenes y utilerías para realizar gráficas en 2D/3D. Xite consiste de programas con interface gráfica de usuario con más de 200 programas que corren en la línea de comandos para PDI. Ambos están disponibles para varias plataformas aunque fueron desarrollados originalmente para el sistema X11/Unix, y sobra decir que son paquetes poderosos, pero demasiado grandes y complejas. Khoros, necesita más de 500MB en disco duro y se recomienda 32MB en memoria 32 más 64MB mínimo de memoria en disco (swap). Xite requiere cerca de 20MB en disco duro una vez instalado.

Al contrario de Xite y Khoros, la biblioteca presentada en este trabajo, *scimagen*, fue desarrollada pensando en mantener una biblioteca lo más pequeña posible y de bajo nivel. Esto permitiría empotrar fácilmente el código en realizaciones *ad hoc* de procesamiento de imagen.

Para mantener el tamaño pequeño de *scimagen* (lea-se como “ese ce imagen”), se usan los programas de visualización de imagen que vienen incluidos en cualquier distribución de GNU/Linux [3], como son los paquetes ImageMagick, xv y gimp [4]. Además, el soporte para formatos de imagen en *scimagen* se limita a solo dos: TIFF y PNM (pbm y ppm). La conversión a otros archivos puede hacerse usando los paquetes anteriores o con NetPBM (que también viene incluido en GNU/Linux). TIFF y PNM son dos formatos portables entre distintas plataformas (Mcintosh, Window y Unix).

PDI puede verse como un conjunto de herramientas y la reutilización de éstas puede resolver un problema dado. Pero el orden de utilización de las herramientas no es siempre el mismo, depende de la aplicación. Esto ha hecho que no se agote la aplicación de PDI a la resolución de problemas y, para resolverlos mejor es necesario conocer el mayor número de las herramientas disponibles y/o modificar ligeramente las existentes.

Las funciones básicas actuales de *scimagen* son: rotación, espejeo (rotación a $\pm 90^\circ$ y $\pm 180^\circ$), cálculo

* Autor a quien dirigir comentarios: fraga@cs.cinvestav.mx

lo del histograma, convolución espacial (detección de bordes, filtro levanta altas (high boost), etc.), normalización del histograma, igualación del histograma y umbralización global.

Scimagen presenta las transformaciones de Fourier, la discreta coseno y la Hartley. Y las aplicaciones en filtrado en frecuencia, tipos pasa-bajas y pasa-altas con perfiles: ideal, coseno alzado, Gauss, Fermi y Butterworth. Filtrado homomórfico y creación de la imagen logaritmo, para visualización del espectro, por ejemplo.

Para el análisis de imagen, en este trabajo se presenta un nuevo algoritmo, y su realización, para la extracción de objetos en imágenes binarias. Este algoritmo es una modificación del descrito por Pavlidis [5]. *Scimagen* cuenta con una rutina para el adelgazamiento binario descrita en [6, p. 67]. Estas dos rutinas nos ayudan en el análisis de imagen, la primera al proveernos de los componentes de una imagen, y la segunda, al darnos otra una representación de los objetos por medio de su esqueleto.

Para la enseñanza de PDI se construyó una interfaz gráfica para la visualización rápida de lo que las operaciones más comunes de PDI pueden realizar. La interfaz gráfica fue realizada con el sistema de desarrollo rápido de interfaces de usuario Qt [7].

Scimagen ha sido desarrollado en el lenguaje C. En C++ podría verse como un solo objeto y cada herramienta como un método disponible de ese objeto. Esta fue la aproximación usada para el desarrollo de la interfaz gráfica.

Se explicarán en detalle las funciones más especiales de *scimagen*: la transformada discreta coseno, la transformada de Hartley, los distintos perfiles usados en el filtraje en frecuencia, su interfaz gráfica y la descripción del algoritmo de extracción de formas.

2. La transformada discreta coseno

La transformada discreta coseno (TDC) es usada para compresión de imágenes debido a que concentra mejor la energía en comparación a otras transformadas. El concentramiento de energía puede verse como sigue: al truncar el número de coeficientes en el dominio transformado, solo son necesarios unos pocos coeficientes para poder recuperar significativamente la información original.

El par de transformadas coseno discretas bidimen-

sionales está definido como:

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (1)$$

para $u, v = 0, 1, 2, \dots, N-1$, y

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u, v) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (2)$$

para $x, y = 0, 1, 2, \dots, N-1$.

La Ec. (1) es la transformada directa y la Ec. (2) es la inversa. Para ambas transformadas los coeficientes son:

$$\alpha(0) = \sqrt{\frac{1}{N}} \text{ y } \alpha(u) = \sqrt{\frac{2}{N}} \text{ para } u = 1, 2, \dots, N-1 \quad (3)$$

La TDC es usada en la compresión de imágenes, especialmente en el formato JPEG [8], es por ello que ha sido programa en Scimagen. Tal vez una utilidad más de la TDC es que se puede visualizar fácilmente su forma separada. Reagrupando la Ec. (1) de la siguiente manera:

$$C(u, v) = \alpha(u) \sum_{x=0}^{N-1} \cos \left[\frac{(2x+1)u\pi}{2N} \right] \alpha(v) \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2y+1)v\pi}{2N} \right] \quad (4)$$

donde la segunda parte de la ecuación forma una TDC unidimensional que puede escribirse como $C(x, v)$ quedando la Ec. (4) como:

$$C(u, v) = \alpha(u) \sum_{x=0}^{N-1} C(x, v) \cos \left[\frac{(2x+1)u\pi}{2N} \right] \quad (5)$$

La Ec. (5) es de nuevo una transformada unidimensional.

De este breve análisis, puede verse que la TDC bidimensional de la Ec. (1) puede evaluarse por TDC unidimensionales sucesivas, como se demuestra en las Ecs. (4) y (5). Esto es, para una

Publicado en el CIE 2002
 imagen se realiza la TDC de las columnas y luego de los renglones (o al revés también es válido, primero los renglones y luego las columnas). La TDC programada en Scimagen sigue la Ec. (5) y es de complejidad N^2 . Existen algoritmos rápidos para evaluar la TDC en un tiempo proporcional a $N \log(N)$ y ya se ha demostrado que todas las transformadas cuyas funciones básicas son funciones senos y cosenos (Fourier, Coseno, Seno, Hartley, etc.) pueden evaluarse con un mismo algoritmo [9].

3. La transformada de Hartley

En 1942, Hartley propuso una transformada y su inversa para realizar análisis espectral de funciones. Su transformada es una alternativa a la transformada de Fourier, útil cuando las funciones que se desea analizar son reales, puesto que la transformada de Hartley de una función real es también real, mientras que la transformada de Fourier de una función real es compleja.

La transformada propuesta por Hartley fue poco atendida, hasta que, en 1983, Bracewell [10] propuso una versión discreta del par de transformadas de Hartley. Desde ese momento se han puesto en evidencia las ventajas que tiene esta transformada para el procesamiento digital de señales y de imágenes, cuando las secuencias de entrada no son complejas. También se han propuesto algoritmos para transformadas rápidas de Hartley [11, 12, 13], que en muchos casos han probado ser más eficientes que las transformadas rápidas de Fourier.

Hartley definió su par de transformadas como:

$$H(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(t) \text{cas}(\omega t) dt$$

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} H(\omega) \text{cas}(\omega t) d\omega$$

donde la función $\text{cas}(a)$ es definida por él mismo como:

$$\text{cas}(a) = \cos(a) + \sin(a)$$

La primera característica importante que debemos notar, es que la transformada directa y la inversa son iguales. Una segunda característica es que la transformada de Hartley de una función $f(t)$ está relacionada con la transformada de Fourier de la forma siguiente:

$$H(\omega) = \Re \{F(\omega)\} - \Im \{F(\omega)\}$$

$$\Re \{F(\omega)\} = [H(\omega) + H(-\omega)]/2$$

$$\Im \{F(\omega)\} = [H(-\omega) - H(\omega)]/2$$

donde $H(\omega)$ y $F(\omega)$ son las transformadas de Hartley y de Fourier, respectivamente, de la misma función $f(t)$; los símbolos \Re y \Im representan la parte real y la parte imaginaria, respectivamente, de un número complejo.

Si $f(t)$ es una función real, entonces en la transformada de Hartley ahorramos las redundancias que se tienen en la transformada de Fourier (la parte real de la transformada de Fourier en una función par, mientras que la parte imaginaria en una función impar).

Las mayoría de las propiedades de la transformada de Hartley [14] son compartidas con la transformada de Fourier [15, 6], ampliamente conocidas; sin embargo, algunos cálculos se hacen más simples por tratar con cantidades reales.

Entre los cálculos que se pueden ver beneficiados, está el de la convolución. El teorema de convolución en el dominio de Hartley puede ser derivado directamente de su relación con la transformada de Fourier, y el teorema de convolución en el dominio de Fourier:

$$2H(\omega) = H_1(\omega) [H_2(\omega) + H_2(-\omega)] + H_1(-\omega) [H_2(\omega) - H_2(-\omega)]$$

La expresión es más larga, pero se debe tener en cuenta que todas las multiplicaciones y sumas son reales, y no complejas como en el caso de la transformada de Fourier. Evidentemente, los cálculos de la convolución pueden ser agilizados

Si una de las transformadas es impar o, como sucede más a menudo en la práctica, si una transformada es par (como los perfiles de un filtro). Si una de las funciones es par, la convolución se simplifica a:

$$H(\omega) = H_1(\omega)H_2(\omega)$$

y esta fórmula es la usada en *Scimagen* para la realización de filtros en imágenes.

4. Perfiles de filtros en frecuencia

El filtraje en el dominio de la frecuencia es una herramienta común en PDI. Un uso es el realzado de imagen para quitar ruido o aislar características importantes en una imagen.

En aplicaciones de biología o microscopía electrónica, las imágenes contienen demasiado ruido. La técnica para mejorar el nivel de la señal de conoce como *promediado de imagen* [6]. En esta se promedia un número dado de imágenes, lo que reduce la varianza del ruido de fondo. La medición de que tanta información se presenta en el promediado final, la resolución, está definido en el dominio de la frecuencia [16]. Teniendo la medida de la resolución de la imagen media final se aplica un filtro pasa bajas para estar seguro que se tiene la información significativa.

Para la aplicación de un filtro en el dominio de la frecuencia existen varias funciones de perfiles posibles. A continuación se describen los posibles perfiles que pueden ser usados y se hará una prueba para ver cual es el mejor de ellos.

En todas las fórmulas siguientes, c es el punto de corte del filtro, o la medida de resolución en el ejemplo presentado anteriormente, r_1 y r_2 son los valores de umbral, r_1 para la banda de paso y r_2 para la banda de corte, como se muestra en la Fig. 1. Todas las fórmulas expresan el caso de un filtro unidimensional, el cambio al perfil bidimensional es directo. Una heurística para hallar los valores para r_1 y r_2 a partir del valor c es:

$$\begin{aligned} r_1 &= c - M/32 \\ r_2 &= c + M/32 \end{aligned} \quad (6)$$

donde M es el tamaño de la secuencia.

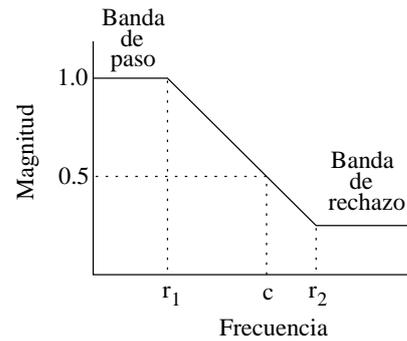


Figura 1: Especificaciones para un filtro en el dominio de la frecuencia. r_1 indica la zona de la banda de paso, r_2 la zona de la banda de rechazo. c es el punto de corte, que se puede situar en cualquier parte dentro de la banda de transición, a 0,5 de la magnitud para la mayoría de los perfiles de filtro aquí usados y a $1/\sqrt{2}$ para los filtros tipo Butterworth

Un filtro pasa bajas *ideal* tiene el perfil:

$$H(u) = \begin{cases} 1, & \text{para } u \leq c \\ 0, & \text{para } u > c \end{cases} \quad (7)$$

El perfil *coseno alzado* usa una función coseno entre 0 y π y sumando una constante de valor 1, para tener un perfil suave en vez del cambio abrupto del filtro ideal. El punto de corte queda al valor 0,5 de la amplitud de la señal de entrada. El perfil queda definido como:

$$H(u) = \begin{cases} 1, & \text{si } u \leq r_1 \\ \frac{1}{2} + \frac{1}{2} \cos \left[\frac{\pi(u-r_1)}{r_2-r_1} \right], & \text{si } u < r_2 \\ 0, & \text{de otro modo} \end{cases} \quad (8)$$

Una función Gaussiana también puede usarse para realizar un perfil suave del filtro, su ecuación queda como:

$$H(u) = \begin{cases} 1, & \text{para } u \leq r_1 \\ \exp \left[- \left(\frac{u-\mu}{\sigma} \right)^2 \right], & \text{de otro modo} \end{cases} \quad (9)$$

donde μ se hace igual a r_1 y $\sigma = (1/k)(c - r_1)$; k es el valor u en el cual e^{-u^2} es igual a 0,5 (para situar el punto de corte a 0,5 de la señal).

También se usa la función Fermi para especificar un perfil:

$$H(u) = \frac{1}{1 + \exp \left(\frac{u-c}{t} \right)} \quad (10)$$

donde t es la temperatura, que puede tomar valores entre 0,002 y 0,1, aquí se usó la heurística $t = 0,01M$ (M es el tamaño de la secuencia). Para un filtro pasa altas tipo Fermi, solo es necesario cambiar el signo de t de la Ec. (10) a negativo.

Los filtro tipo Butterworth resultan de la ecuación

$$|H(u)| = \frac{1}{\sqrt{1 + (u/c)^{2N}}} \quad (11)$$

donde N es el orden del filtro. Este es uno de los filtros más comúnmente usados y está totalmente documentado en la literatura [15, p. 703], al contrario de los otros perfiles que su existencia, creemos, se debe a la facilidad con que se pueden definir nuevas funciones de perfiles.

En la Fig. 2 pueden visualizarse los distintos perfiles de la Ecs. (7), (8), (9), (10) y para N igual a 2, 3 y 8 para la (11), todos para un punto de corte normalizado de 0,25.

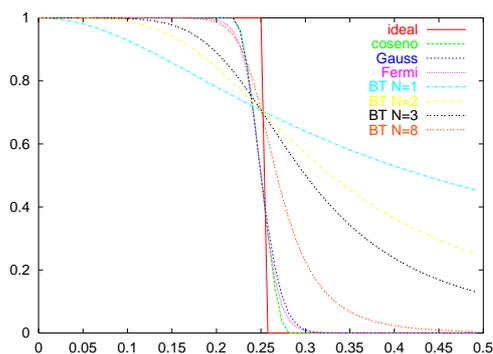


Figura 2: Forma de los distintos perfiles para un filtro en el dominio de la frecuencia. El punto de corte usado para todos los filtros es de 0,25. Sólo se presenta la mitad de las frecuencias debido a que la otra mitad es simétrica a éstas.

La efectividad de cada perfil puede medirse viendo la respuesta al escalón unitario [15, p. 115] de cada perfil, esto es, su respuesta en el tiempo, en 1D, o en el espacio, en 2D. La respuesta al escalón unitario para una secuencia está dada por:

$$s[n] = \sum_{m=-\infty}^n h[m] \quad (12)$$

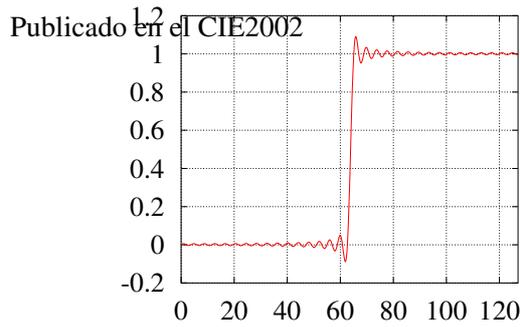
donde $h[m]$ es la respuesta al impulso unitario, y ésta se obtiene simplemente realizando la transformada de Fourier inversa de cada una de las

ecuaciones (7), (8), (9), (10) y (11), ó que es lo mismo, de las formas mostradas en la Fig. 2. En la Fig. 3 se puede comparar visualmente el comportamiento en el dominio del tiempo de cada uno de los perfiles. Las distorsiones que se ven en la Fig. 3(a) se conocen como *fenómeno de Gibbs* [15, p. 200] y se deben al truncamiento de las frecuencias debido al filtro ideal [Ec. (7)]. Las distorsiones debidas a los perfiles coseno alzado, Gauss y Fermi son muy semejantes, prácticamente indistinguibles y decrecen en cuatro ciclos con respecto al perfil ideal, como se ven en la gráfica de la Fig. 3(b). Las distorsiones de los perfiles tipo Butterworth son mejores respecto a los anteriores a que las oscilaciones decrecen en la mitad de tiempo, en dos ciclos con respecto al perfil ideal. Sin embargo el pico máximo del perfil Butterworth de orden 8 es casi igual al del perfil ideal. Claramente el mejor filtro, el que produce menos distorsiones, es el Butterworth de orden 2.

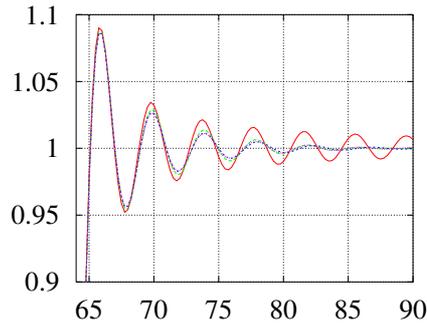
En la Fig. 4 pueden verse las distorsiones en 2D producidas por los distintos filtros. Claramente la mayor distorsión se produce aplicando el filtro ideal (Fig. 4(b)). La distorsión es semejante con los perfiles coseno alzado (Fig. 4(c)), Gauss (Fig. 4(d)) y Fermi (Fig. 4(e)). El filtro Butterworth de orden 8 (Fig. 4(h)) produce casi tanta distorsión que los perfiles anteriores. Si queremos reducir las distorsiones tenemos que extender los perfiles no haciéndolos tan abruptos, como se ven en la Fig. 2 y haciéndolos más parecidos a un Butterworth de orden 2 ó 3 (Figs. 4(f) y 4(g)), pero mezclamos al mismo tiempo todas la frecuencias. Es un compromiso que debe hacerse. De estos resultados creemos que el perfil que tiene un buen comportamiento es el Butterworth de orden 3.

5. Restauración con el filtro de Wiener

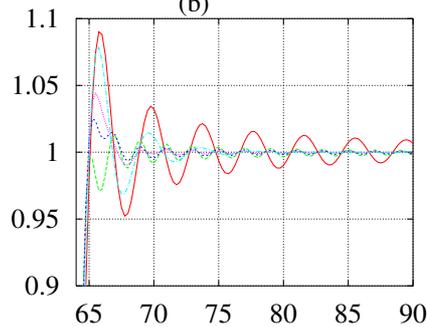
Dentro de los conceptos básicos de PDI se encuentra la *restauración de imagen*, que puede verse como el proceso mediante el cual se reconstruye una imagen $f(x, y)$ que ha sido degradada. La técnica está basada en modelar matemáticamente el fenómeno de degradación de la imagen y aplicar el proceso inverso para recobrar la imagen ori-



(a)



(b)



(c)

Figura 3: Respuesta al escalón unitario de un filtro pasa-bajas a la frecuencia normalizada de 0.25, en (a) del filtro pasa bajas ideal. Las gráficas (b) y (c) son acercamientos de la misma gráfica en (a) y comparan, (b) los perfiles coseno alzado, Gauss y Fermi con el ideal, y (c) los perfiles Butterworth de orden 2, 3 y 8 con el ideal.

ginal. Desde el punto de vista psicovisual, tiene esta técnica el impacto más grande, sin embargo está limitada por la modelación efectiva de la perturbación.

El modelo general del sistema es:

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y) \quad (13)$$

donde $g(x, y)$ es la imagen de salida, $f(x, y)$ es la imagen de entrada, $h(x, y)$ es la perturbación

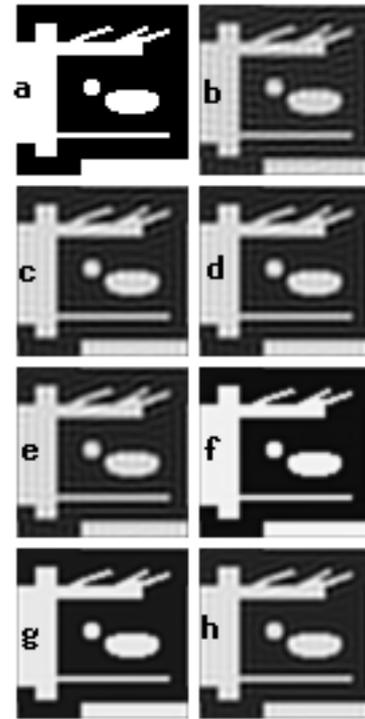


Figura 4: Efectos de los diversos perfiles en una imagen. (a) Imagen original de 64×64 pixels, (b)-(h) imágenes de resultado de aplicar un filtro pasa bajas a la frecuencia normalizada de 0.25 con los perfiles (b) ideal, (c) coseno alzado, (d) Gauss, (e) Fermi, (f) Butterworth (BT) orden 2, (g) BT orden 3 y (h) BT de 8.

aplicada, el operador $*$ representa la convolución entre $h(x, y)$ y $f(x, y)$, y $\eta(x, y)$ una señal adicional de entrada que puede verse como una interferencia o ruido.

Considerando el sistema de la Ec. (13) como lineal, al aplicarle la transformada de Fourier obtenemos:

$$G(u, v) = H(u, v)F(u, v) + N(u, v) \quad (14)$$

donde cada término está ahora en el dominio de la frecuencia (u, v) y podemos despejar F obteniendo:

$$F(u, v) = [G(u, v) - N(u, v)] / H(u, v) \quad (15)$$

Si el ruido lo consideramos igual a cero, podemos estimar nuestra imagen de entrada simplemente como $\hat{F} \approx G/H$. Sin embargo se nos presenta el problema de que la función H presenta muchos ceros y no podemos realizar la división.

La matriz H puede verse un *filtro* y el método se conoce como *filtrado inverso* de imagen. El filtro de Wiener realiza la siguiente aproximación [6, p. 280]:

$$F(\hat{u}, v) \approx \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \right] G(u, v) \\ \approx \left[\frac{H^*(u, v)}{|H(u, v)|^2 + K} \right] G(u, v) \quad (16)$$

La Ec. (16) se obtiene considerando $|H(u, v)|^2 = H^*(u, v)H(u, v)$.

Para probar el filtro de Wiener se realizó lo siguiente: 1) A una imagen dada se le aplica una de las distorsiones ofrecidas por el programa *gimp* [4]. 2) Se creó una imagen con un solo punto blanco central, lo que forma una imagen impulso unitario. 3) Se genera una nueva imagen aplicando la misma distorsión usada en el paso (1) a la imagen del paso (2) y así modelizamos la respuesta al impulso unitario de la distorsión producida por *gimp*. 4) Tomando la Transformada de Fourier (TF) de la imagen anterior tenemos el término $H(u, v)$ de la Ec. (16). 5) Finalmente aplicamos el filtro de Wiener dado por la Ec. (16) usando la TF de la imagen inicial, tenemos la imagen final restaurada. El valor de K del filtro tiene que sintonizarse por prueba y error. Un ejemplo de la aplicación de este procedimiento puede verse en la Fig. 5

La efectividad de la técnica mostrada en la Fig. 5 está limitada por la facilidad de contar con la respuesta al impulso unitario. En este caso ha sido muy fácil como se ve en el paso (2) y (3), pero en ciertas aplicaciones, como medir la respuesta de un microscopio electrónico, es imposible de hacer. En estos casos la respuesta al impulso unitario tiene que ser estimada por otros medios.

6. La interfaz de usuario

En la Fig. 8 se observa el comportamiento de la interfaz gráfica de *Scimagen*. La filosofía de construcción de la interfaz ha sido la de un programa tipo UNIX: al llamar al programa se despliega una imagen de presentación, no se abre un menú ni iconos, al hacer clic sobre la imagen de presentación se abre un menú que nos permite abrir una

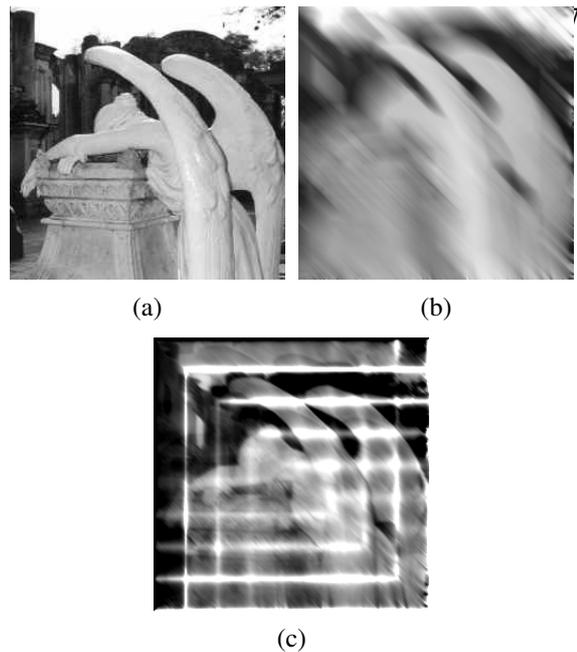


Figura 5: Restauración de imagen usando el filtro de Wiener. En (a) la imagen original; (b) la imagen en (a) distorsionada con movimiento a 45° con una “fuerza” de 30, usando el programa *gimp*; (c) la imagen (b) restaurada, se usó $K = 0,005$ para la Ec. (16)

imagen y realizar las operaciones más comunes de procesamiento de imagen, tales como rotar la imagen, aplicarle filtros espaciales y en el dominio de la frecuencia, invertir el contraste, normalizar e igualar la imagen y sacar su histograma. El histograma se despliega con el uso del programa *gnuplot*. Así, en la Fig. 8 se ha abierto la imagen (1), después de le ha aplicado un suavizado (un filtro pasa bajas espacial, donde el valor de cada pixel es sustituido por el promedio de el mismo y sus ocho pixels vecinos) y el resultado se despliega en la ventana (2) y a ésta imagen suavizada se la ha invertido el contraste y el resultado se presenta en una nueva ventana, la (3) en la misma Fig. 8. También se visualiza parte del menú con las opciones para el rotado de imagen; el menú se ha abierto al hacer clic sobre la imagen invertida (3). Esta interfaz gráfica se desarrolló con fines didácticos para apreciar la facilidad del diseño de ella con la biblioteca Qt [7] y el diseño de todas las herramientas de procesamiento de imagen de *Scimagen* en un solo objeto de C++, y la filosofía de diseño de una interfaz para un sistema Unix

8 GNU/Linux. Así mismo, la interfaz gráfica no fue diseñada para facilitar el uso de las herramientas de *Scimagen* sino dar a conocer las herramientas más comunes de procesamiento de imagen.

7. Algoritmo para la extracción de formas

Partiendo de la misma idea de Pavlidis, una forma, o una figura compuesta por pixels de un mismo tono, se considera compuesta por “tramos”. Un tramo es una secuencia de pixels de un mismo tono definido por su punto extremo más a la izquierda, como puede verse en la Fig. 6



Figura 6: Un tramo. El punto indica el pixel más a la izquierda.

Cada tramo puede estar conectado con otros arriba y abajo de él. Los casos posibles de conectividad entre tramos vienen indicados en la Fig. 7.

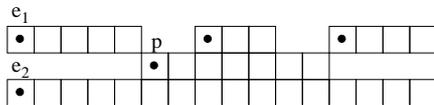


Figura 7: Los cuatro posibles casos de conectividad para un tramo dado p .

Existe una función, que llamaremos *ligas()*, que para un tramo dado p (ver la Fig. 7) debe de realizar lo siguiente:

- Cuenta el número de tramos arriba y abajo de p .
- Si en cualquiera hay más de un tramo, se ponen en una pila. La función regresa los dos puntos extremos de los tramos extremos, esto es, los puntos e_1 y e_2 marcados en la Fig. 7.

La extracción de una forma la realizamos rellenándola en otro tono. El algoritmo lo inicializamos con cualquier pixel que conforme la forma y llamando a una función *izquierda()* que nos regrese su pixel más a la izquierda. Este pixel lo metemos en la pila.

Finalmente el algoritmo queda como:

```

dirección = ARRIBA;
while( pop( p ) ) {
    while ( 1 ) {
        ligar( p );
        if ( e1 == 0 AND e2 == 0 ) {
            if ( el tramo no está rellenado ) re-
                llenar( p );
            break;
        }
        if ( e1 == 0 ) direccion=ABAJO;
        if ( e2 == 0 ) direccion=ARRIBA;
        if ( e1 > 0 AND e2 > 0 ) {
            if ( direccion==ARRIBA ) push(e2);
            else push( e1 );
        }
        rellenar( p );
        p = e_direccion
    }
}

```

La función *rellenar(p)* cambia el tono de la forma. Así, si tenemos una imagen en blanco y negro, debemos de usar esta función asignando el blanco al valor 255, esto es, cambiamos la imagen a 1 byte/pixel. De esta forma la forma que se está extrayendo puede rellenarse con el valor de 10, por ejemplo, y su extracción de la imagen total se realiza con una simple umbralización global.

8. Conclusiones

Se ha presentado una nueva biblioteca de procesamiento digital de imagen (PDI), llamada *Scimagen*, de bajo nivel, desarrollada en lenguaje C, cuyas funciones pueden empotrarse en nuevos desarrollos de PDI. Otra motivación para el desarrollo de esta biblioteca ha sido el contar con una plataforma propia de bajo nivel que consideramos necesaria para la enseñanza de PDI. La posibilidades de *Scimagen* se han presentado en este trabajo con la descripción de las transformada coseno discreta (TDC), la transformada de Hartley (TDH), un estudio de los perfiles usado en el filtraje de imágenes en frecuencia, en una presentación del

Publicado en el CIE2002
filtro de Wiener y la interfaz gráfica de *Scimagen*.
Se ha descrito también un nuevo algoritmo para la extracción de formas de imágenes binarias. La TDC se usa para la compresión de imágenes y el uso de la TDH reduce el costo computacional al ofrecer las mismas facilidades de la transformada de Fourier pero con números reales y no complejos, por ello se tiene disponibles en *Scimagen*. El estudio de los muchos perfiles de filtro que se han usado en el dominio de la frecuencia, resuelve el comportamiento de ellos y se ha llegado a la conclusión que el comportamiento es el mismo para los perfiles coseno alzado, Gauss y Fermi. Con la metodología de la restauración con filtro de Wiener se ha explicado una de las aplicaciones más efectivas de PDI, pero también más difíciles de realizar. *Scimagen* está ahora mismo en continuo desarrollo, con algunas herramientas más que ofrecerá en un futuro, y ofrecerá además algún algoritmo para clasificación de formas y para sacar características representativas de formas.

Agradecimientos

Este trabajo ha sido parcialmente apoyado por el proyecto JIRA2001/07 del CINVESTAV-IPN.

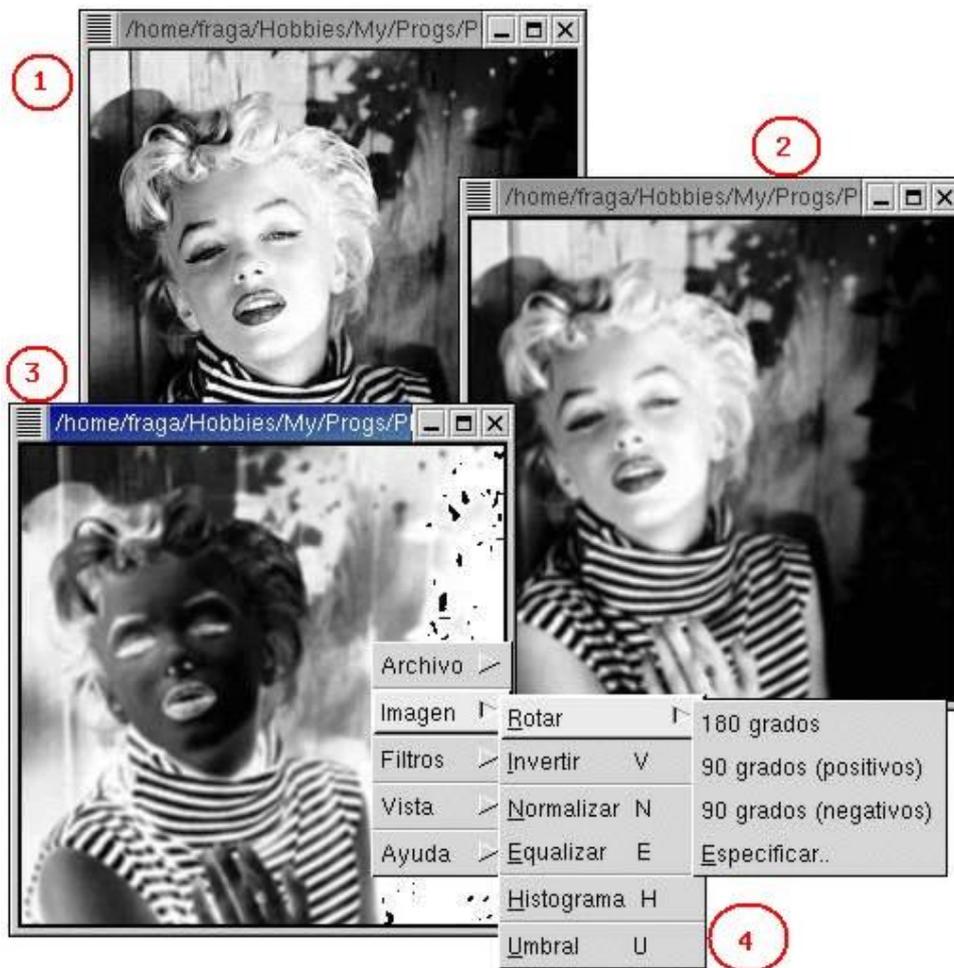


Figura 8: La interfaz de usuario de Scimagen. En (1) la primera imagen abierta, en (2) la imagen (1) con un filtro espacial pasa bajas, y en (3) la imagen (2) con contraste invertido. En (4) se notan el menú que se ha abierto al hacer clic con el ratón sobre la imagen (3)

Referencias

- [1] Xite, image processing laboratory, department of informatics. <http://www.ifi.uio.no/blab/Software/Xite/>.
- [2] J. Rasure and C. Williams. An integrated visual language and software development environment. *Journal of Visual Languages and Computing*, 2:217–246, 1991. <http://www-vis.lbl.gov/software/khoros.html>.
- [3] R. Stallman. Linux and the gnu project. <http://www.gnu.org/gnu/linux-and-gnu.html>.
- [4] L.G. de la Fraga. Xv, ImageMagick y gimp, tres programas para la visualización y edición de imágenes. *Included in CIE2002*, 2002.
- [5] T. Pavlidis. *Algorithms for Graphics and Image Processing*. Computer Science Press, 1982.
- [6] Gonzalez R.C and R.E Woods. *Digital Image Processing*. Adisson-Wesley, 1992.
- [7] Qt, the cross-platform c++ gui toolkit. <http://www.trolltech.com>.
- [8] G.K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44, Apr 1991.
- [9] V. Britanak V and K.R. Rao. The fast generalized discrete fourier transforms: A unified approach to the discrete sinusoidal transforms computation. *Signal Processing*, 79(2):135–150, 1999.
- [10] R.N. Bracewell. Discrete Hartley transform. *Journal of the Optical Society of America*, 73(12):1832–1835, Dec 1983.
- [11] R.N. Bracewell. The fast Hartley transform. *Proceedings of the IEEE*, 72(8):1010–1018, Aug 1984.
- [12] R.F. Ullman. An algorithm for the fast Hartley transform. Technical Report 38, Stanford Exploration Project (SEP), 1984. <http://sepwww.stanford.edu/oldreports/sep38/index.html>.
- [13] G.A. Bi and S.T. Lian. Fast algorithms for generalized discrete Hartley transform. *Journal of Circuits Systems and Computers*, 10(1-2):77–83, 2000.
- [14] R. Otolini. Migration by Hartley transform. Technical Report 38, Stanford Exploration Project (SEP), 1984. <http://sepwww.stanford.edu/oldreports/sep38/index.html>.
- [15] A.V. Oppenheim, A.S. Willsky, and S.H. Nawab. *Signals and Systems*. Prentice Hall, 2nd edition, 1997.
- [16] L.G. de la Fraga, J. Dopazo, and J. M. Carazo. Confidence limits for resolution estimation in image averaging by random subsampling. *Ultramicroscopy*, 60:385–391, 1995.