

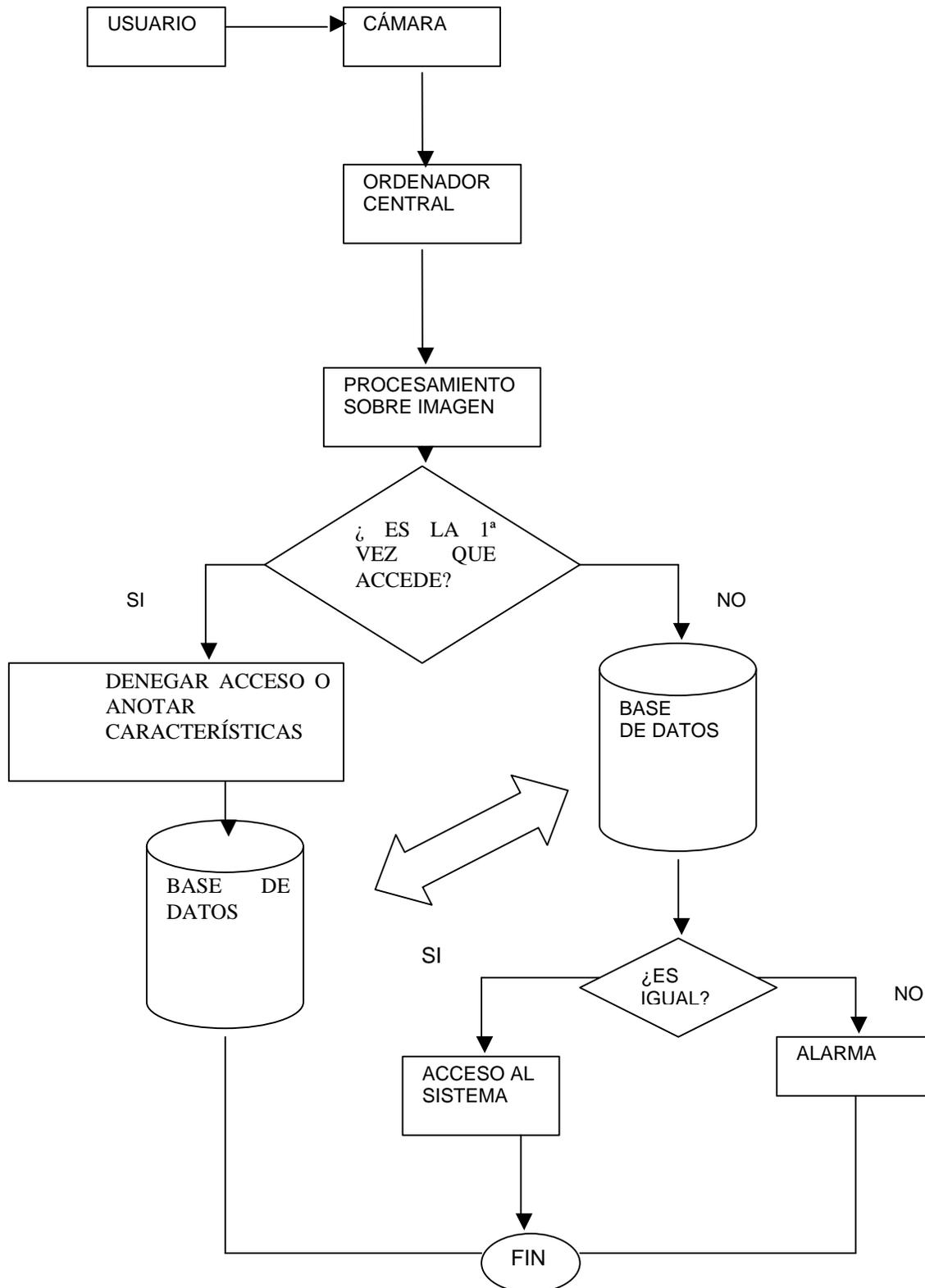
PRIMERA PARTE. INTRODUCCIÓN

Ante un mundo tan globalizado como en el que nos encontramos, en el que todos quieren interactuar con todos y como consecuencia del auge de las tecnologías de la información ha surgido la necesidad de privacidad, en el sentido que nadie entre en el entorno diseñado sin alguna autorización. Como consecuencia de esta necesidad se intenta mejorar la seguridad en el acceso a los sistemas de información. Es por ello que las empresas y nombro empresas como podría nombrar cualquier otra organización, intentan mejorar su seguridad para evitar el intrusismo en las mismas y que la competencia pueda conocer sus datos estratégicos del negocio. Por todo ello en sus intranets conectan Firewall para intentar que cualquier persona ajena a la organización pueda acceder. En sus intranets corporativas se accede mediante login y password pero esto no es suficiente porque no sabemos realmente quien es la persona física que está detrás de ese login y password. Puede ser la persona que realmente tiene acceso a través de esa contraseña o puede ser la persona que se lo ha robado. Por todo ello y en entornos que requieren de máxima seguridad es necesario no sólo que el usuario introduzca su login y contraseña sino el reconocimiento físico de la persona que intenta acceder a ese servicio. Para ello cada usuario debe disponer en su puesto de acceso una cámara en la cual se recoja su imagen y mediante un procesamiento sobre la misma que nos permita detectar determinadas características de ella poder emitir el juicio que la persona que accede a través de ese login y password es quien dice ser.

Las motivaciones que nos ha llevado a plantear este estudio es que, si bien con la tecnología que disponíamos no nos podíamos permitir estas formas de reconocimiento de personas y utilizamos la firma, la huella dactilar o cualquier otra forma, en ambientes controlados que requieran de una condiciones extremas de seguridad sería de gran utilidad plantearse los métodos de reconocimiento de contornos en cara que permitirán anotar las características de la misma en una base de datos. Una vez que se tienen estas características almacenadas en la base de datos cuando una persona se presenta ante la cámara se recoge la imagen se realiza el procesado sobre la misma previamente determinado y con las características de este procesado se va a la base da datos para comprobar que las características coinciden. Si coincide se le permitirá el acceso al sistema y si no dependiendo del entorno en el que se encuentre se le denegará el acceso y saltará una alarma por intento de intrusismo.

Las aplicaciones que puede tener este sistema son la siguientes:

- Evitar el acceso de personal no autorizado al CSIC, organismos públicos, bancos, cámaras de seguridad.
- Se podría utilizar no sólo para detectar sólo la cara sino para comprobar que la persona que accede a determinado sector cumple con los requisitos de uniformidad exigidos por la normativa ya establecida.
- Para evitar que cualquier hacker o pirata informático pueda acceder a información de la empresa.



SEGUNDA PARTE. FUNDAMENTO TEÓRICO

1. INTRODUCCIÓN

En un proceso de visión artificial, el procesamiento de la imagen tiene como finalidad producir otra imagen de mayor calidad que simplifique y facilite etapas posteriores. El objetivo, por tanto, no es extraer información de la imagen sino actuar convenientemente sobre los niveles de gris de ella para compensar defectos de iluminación (poco contraste, no-uniformidad, etc) así como eliminar ruido y defectos espurios. En un ámbito más global, el procesamiento de imágenes no se limita a estas operaciones sino que incluye además la compresión y descompresión de imágenes, restauración, etc. El campo de aplicaciones es muy extenso incluyendo aplicaciones médicas, comunicación de vídeo, meteorología, codificación, etc.

Históricamente, el procesamiento de imágenes se ha desarrollado desde dos enfoques diferentes. Uno ha sido la extensión natural del procesamiento digital de señales, desde donde se ha abordado como un problema de teoría de filtrado de señales, involucrando de manera importante operaciones en el dominio de la frecuencia. El segundo enfoque, más heurístico, considera la imagen digital como una matriz de muestras discretas, realizando operaciones aritméticas sobre los elementos de esta matriz. En este trabajo se emplea fundamentalmente el segundo punto de vista, si bien en algunos casos se recurre a interpretaciones propias del enfoque de teoría de la señal.

2. CONCEPTOS BÁSICOS PARA EL PROCESAMIENTO DE IMÁGENES

2.1 HISTOGRAMAS

El histograma de una imagen es una representación gráfica de la frecuencia con la que los niveles de grises aparecen en dicha imagen. El eje de abscisas indica los distintos niveles (discretos) de grises y en ordenadas se representa la frecuencia o, también a veces, el número de píxeles que poseen el nivel de gris (ver figura 1.1). El histograma se construye, simplemente, rastreando toda la imagen y contabilizando el número de píxeles que poseen cada nivel de gris.

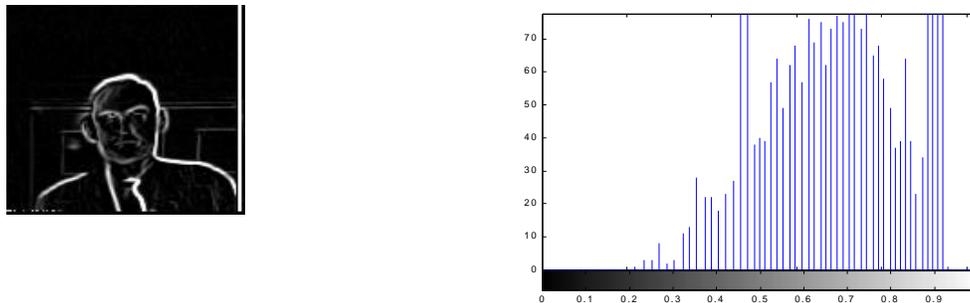
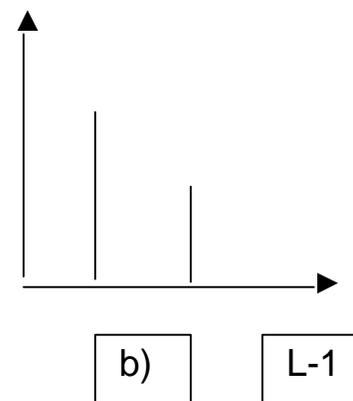
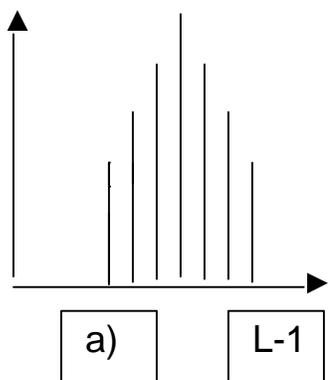


Figura 1.1.- Una imagen con su correspondiente histograma

El histograma proporciona información estadística sobre cómo están distribuidos los distintos niveles de grises de la imagen. Esta información puede ser bastante útil para tareas tales como conocer si la digitalización se ha efectuado correctamente, decidir el valor umbral de una imagen, tener una estimación del brillo medio y contraste, etc. En la figura 1.2 se muestran algunos histogramas típicos donde es posible extraer cierto tipo de información.

Cuando el histograma se expresa en términos de frecuencia de aparición de los distintos niveles de grises es posible establecer una cierta analogía entre éste y una función de densidad de probabilidad (f.d.p.). Lógicamente, esta analogía será más sólida cuanto mayor sea el número de niveles de intensidad de la imagen.



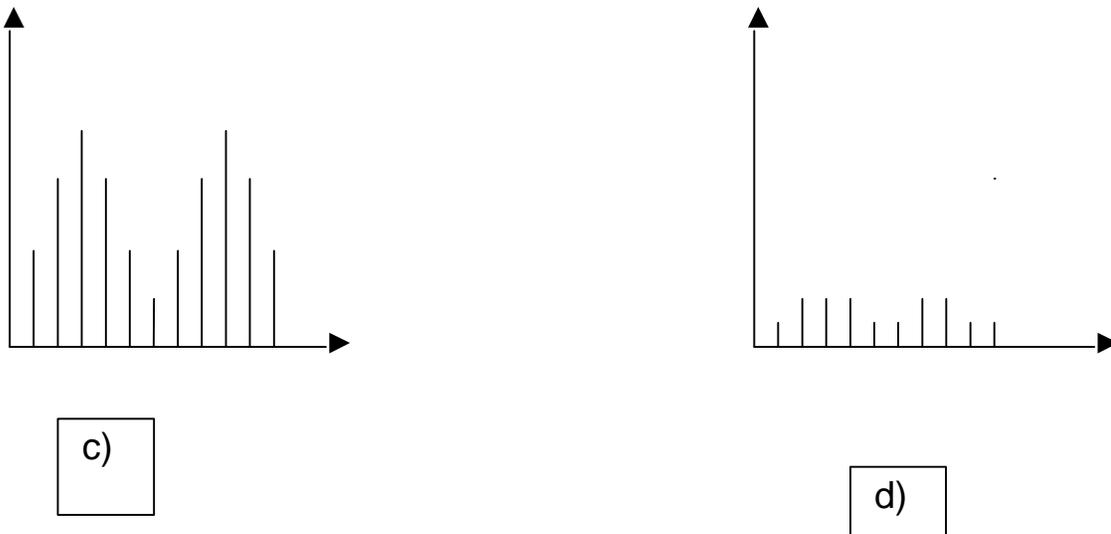


Figura 1.2.- Algunos histogramas típicos. (a)Imagen con mucho brillo y poco contraste (mala digitalización). (b)Imagen binaria con dos niveles A y B. (c) Histograma bimodal típico de objeto sobre fondo. (d) Saturación del blanco y negro (mala digitalización).

2.2 TABLAS DE CONSULTA

Una tabla de consulta, normalmente conocida como LUT (look-up-table), es una función discreta definida sobre los números naturales comprendidos en el intervalo $[0, L-1]$, siendo L el número de niveles de grises empleados en la digitalización. Esta función define una transformación píxel a píxel entre los niveles de grises de la imagen a procesar y de la imagen resultante. Normalmente, el "0" corresponde al negro o nivel más oscuro y el $L-1$ al blanco o nivel de gris más claro.

En la figura 1.3 se ilustra el principio de funcionamiento de una LUT. En la práctica, puesto que los niveles son discretos, la transformada se almacena en una tabla (vector) cuyos elementos son los nuevos niveles de grises correspondientes al intervalo $[0, L-1]$. La transformación, por tanto, consiste en asignar como nuevo nivel de gris el correspondiente al elemento indexado con el nivel de entrada.

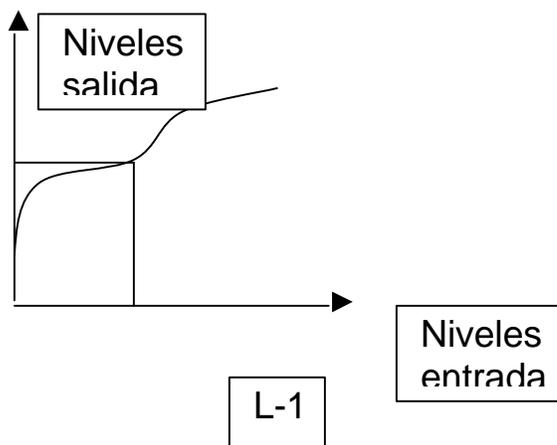


Figura 1.3- *Funcionamiento de una tabla de consulta (LUT).(a) Transformación continua.(b) Transformación discreta.*

Este tipo de transformación, aunque muy simple, tiene una gran utilidad en el procesamiento de imágenes, ya que su implementación es fácil y, sobre todo, se puede aplicar en tiempo real (a la frecuencia de digitalización). Aunque posteriormente se propondrán varias LUT's en la **figura 1.4** se exponen algunos ejemplos a modo de ilustración.

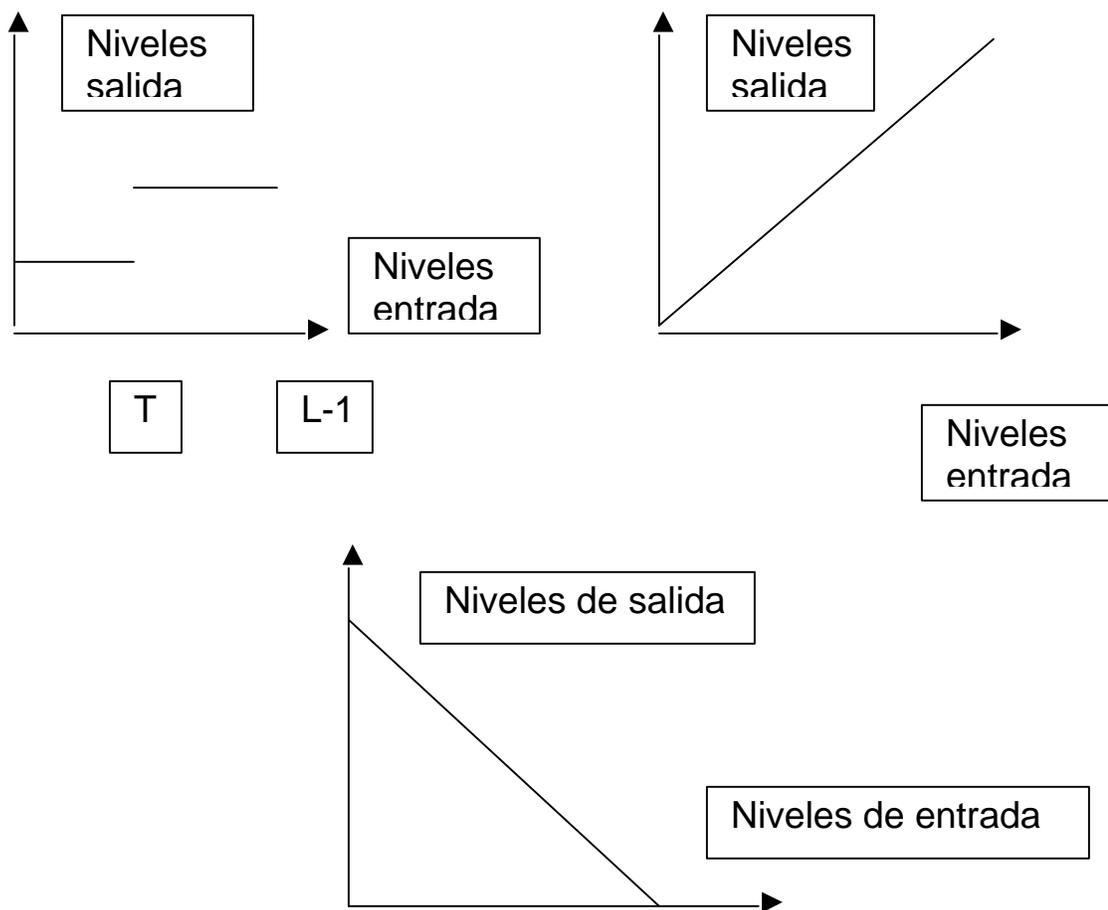


Figura 1.4.- Algunos ejemplos simples de tablas de consulta (LUT). (a) Operación de umbralización. (b) No se modifica la imagen (operación identidad). (c) Inversión de la paleta de grises.

2.3 RELACIONES BÁSICAS ENTRE PÍXELES

2.3.1 MEDIDAS DE DISTANCIA

Se dice que D es una función de distancia si verifica:

1. $D(p_1, p_2) \geq 0$ [$D(p_1, p_2) = 0$ si $p_1 = p_2$]
2. $D(p_1, p_2) = D(p_2, p_1)$
3. $D(p_1, p_3) \leq D(p_1, p_2) + D(p_2, p_3)$

donde p_1, p_2 y p_3 son tres píxeles de coordenadas $(x_1, y_1), (x_2, y_2)$ y (x_3, y_3) respectivamente.

La distancia euclídea entre p_1, p_2 se define como:

$$D_e(p_1, p_2) = [(x_1 - x_2)^2 + (y_1 - y_2)^2]^{1/2}$$

El lugar geométrico de los píxeles cuya distancia euclídea a “ p_1 ” es menor o igual que un valor dado r es un círculo de radio r centrado sobre “ p_1 ”.

Además de la función de distancia euclídea suelen utilizarse también otras métricas, más adecuadas al formato discreto de las imágenes digitales, entre las que se encuentran la distancia D_r , denominada rectangular o Manhattan, definida como:

$$D_r(p_1, p_2) = \text{abs}(x_1 - x_2) + \text{abs}(y_1 - y_2)$$

y la distancia D_t denominada de Tchebichev, y definida como:

$$Dt(p_1,p_2)=\max[\text{abs} (x_1- x_2)+ \text{abs} (y_1 - y_2)]$$

El lugar geométrico de los píxeles a distancia D_r y D_t de “ p_1 ” menores que un valor dado r son un rombo y un cuadrado, respectivamente, centrados sobre el píxel “ p_1 ” (ver figura 1.5).



Figura 1.5.- *Distancia D_r y D_t al píxel central.*

2.3.2 VECINOS DE UN PÍXEL

Se define el conjunto de los 4-vecinos de un píxel dado “ p ”, denotado por $N_4(p)$, como el conjunto formado por los píxeles de arriba, abajo, a la derecha y a la izquierda de “ p ”, es decir, por aquellos píxeles “ q ” que verifican $D_r(p,q)=1$ (ver **figura 1.6**).

Se define el conjunto de los vecinos diagonales de “ p ” denotado por $N_d(p)$, como el conjunto formado por los píxeles situados en las diagonales de “ p ” y a distancia D_t igual a uno (ver **figura 1.6**).

Finalmente, se define el conjunto de 8-vecinos de “ p ”, denotado por $N_8(p)$, como el conjunto formado por aquellos píxeles “ q ” que verifican $Dt(p,q)=1$ (ver **figura 1.6**).

	X	
X	P	X
	X	

X		X
	P	
X		X

X	X	X
X	P	X
X	X	X

Figura 1.6.- *Diferentes conjuntos de vecinos del pixel “ p ”.*

2.4 OPERACIONES DE CONVOLUCIÓN

La convolución es una operación que modifica el nivel de gris de los píxeles de la imagen teniendo en cuenta los píxeles de su entorno de vecindad. Este tipo de operación, a menudo denominada filtrado de la imagen, es fundamental como herramienta para el procesado de la imagen, y en particular, para la eliminación de ruido y detección de bordes.

El producto de convolución entre dos señales bidimensionales discretas **f** y **g** se define como:

$$\mathbf{f} \otimes \mathbf{g} = \sum_m \sum_n h(m, n) f(i-m, j-n)$$

donde los sumatorios se extienden sobre el área de solapamiento de las dos señales.

En el caso de imágenes digitales, una de las señales, por ejemplo **f**, es la imagen a procesar mientras que la otra **h** es denominada máscara o filtro de convolución. El tamaño de la máscara de convolución puede ser muy variado, desde 3x3 hasta máscaras enormes (por ejemplo 63x63) empleadas en operaciones de suavizado gaussiano.

En la figura 1.7 se ilustra gráficamente el funcionamiento de la operación de convolución. Como se desprende de la ecuación anterior, esta consiste en sumar de forma ponderada, para cada píxel (i,j) de la imagen **f**, los niveles de grises de su entorno de vecindad. La ponderación de cada píxel considerado viene dado por el valor almacenado en la máscara de convolución **h**. El resultado obtenido es el nivel de gris del píxel (i,j) de la imagen resultante **g**.

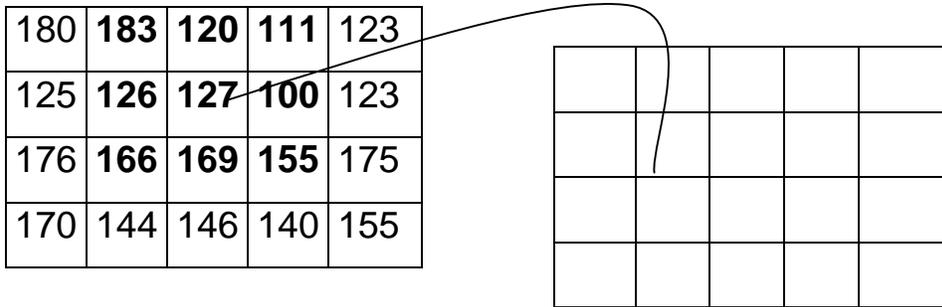


Imagen original $f(i,j)$ Imagen resultante f^g

Figura 1.7.- Ilustración de la operación de convolución.

Hay que hacer notar que la máscara h ha de ser rotada 180° antes de efectuar el producto con los niveles de gris de la imagen f , es decir, $f(i-1,j-1)$ debe multiplicarse por $h(1,1)$, $f(i-1,j)$ por $h(1,0), \dots$, y $f(i+1,j+1)$ por $h(-1,-1)$. Muy a menudo esta rotación no es necesaria puesto que las máscaras son simétricas o antisimétricas y el resultado es tomado en valor absoluto.

3. REALCE

El principal objetivo de las técnicas de realce que se presentan en esta sección es aumentar el contraste mediante la redistribución de los niveles de grises de la imagen. En visión por computador, la finalidad de este aumento de contraste no es mejorar la calidad de la imagen para una mejor visualización, sino obtener una imagen resultante que sea más adecuada que la original para una determinada aplicación.

Una forma simple pero eficaz de aumentar el contraste, bien de una imagen completa o bien de determinadas regiones de interés

dentro de ella, consiste en la modificación de la escala de grises de dicha imagen mediante una tabla de consulta o LUT. A continuación se presentan dos técnicas que utilizan directamente la información proporcionada por el histograma de la imagen para generar la LUT. Posteriormente, se describe la operación de modificación de la escala de grises, para finalizar con el planteamiento de las técnicas de realce local. Salvo en este último caso, las técnicas presentadas son transformaciones píxel a píxel, sin tener en cuenta otros niveles de grises del entorno de vecindad.

3.1 IGUALACIÓN DE HISTOGRAMA

Como ya ha sido comentado, el histograma de una imagen proporciona una descripción global de la apariencia de la imagen. Los métodos de realce que se estudian en este apartado y en los siguientes están basados en la modificación de los niveles de grises de tal forma que la imagen resultante presente un histograma similar a uno previamente especificado.

En concreto, en este apartado se analiza el caso particular en el cual el histograma especificado es uniforme a lo largo de todo el rango de intensidades. En primer lugar, se planteará el caso hipotético en el cual las intensidades de la imagen toman valores continuos y además tienen un comportamiento aleatorio de acuerdo con una determinada distribución de probabilidad.

3.1.1 CASO CONTINUO

Sea “r” la variable que representa la intensidad de los píxeles de la imagen **f** que se desea realzar. Supóngase inicialmente que “r” está normalizada y es continua con valores en el intervalo [0,1]. Para cualquier valor de intensidad “r” en dicho intervalo, se considerarán transformaciones T, tal que:

- a) $0 \leq T(r) \leq 1$ para $0 \leq r \leq 1$
- b) T(r) no está multievaluada.
- c) T(r) es monótona creciente.

Sea “s” la variable de intensidad de la imagen **g**, resultante de dicha transformación ($s=T(r)$). La transformación inversa viene dada por:

$$r = T^{-1}(s) \text{ para } 0 \leq s \leq 1$$

y satisface también las condiciones anteriores con respecto a la variable “s”.

Así, si consideramos los histogramas de $p_f(r)$ y $p_g(s)$ en términos de frecuencia normalizados de la imagen original y transformada respectivamente, podemos establecer cierta analogía entre los histogramas y las funciones de densidad de probabilidad siempre que se suponga un comportamiento aleatorio para las variables “r” y “s”.

Por consiguiente si conozco la función de densidad de probabilidad $p_f (r)$ y la función de transformación $T (r)$ la función densidad de probabilidad de las intensidades transformadas viene dada por:

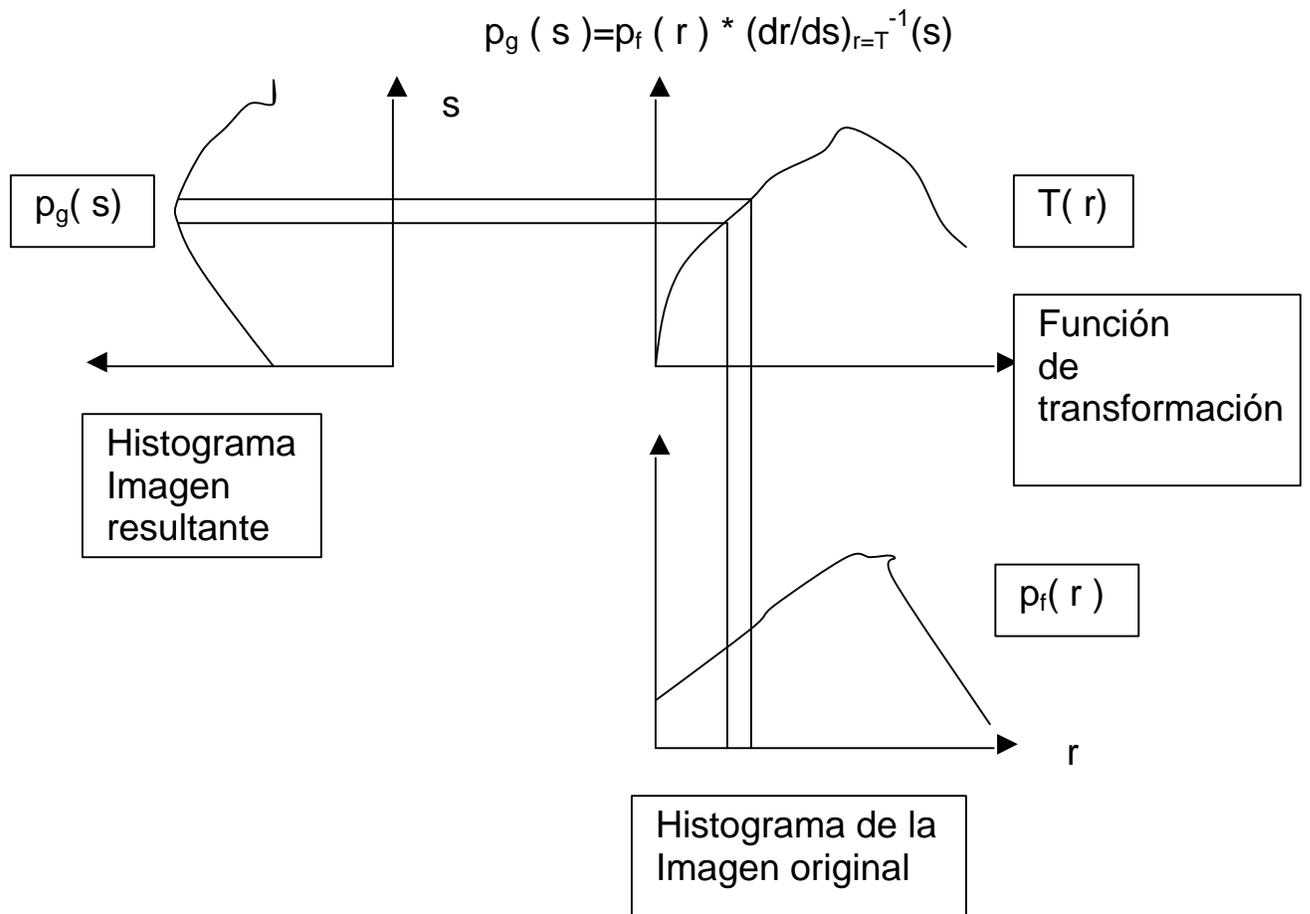


Figura 1.8- *Modificación del histograma (cambio niveles grises)*

Considérese la función de transformación:

$$s = T (r) = \int_0^r p_f (w) dw \text{ para } 0 \leq r \leq 1$$

donde w es la variable auxiliar de integración.

Para esta función de transformación se tiene:

$$(ds/dr=p_f(r)) \Rightarrow (dr/ds(r)=1/p_f(r))$$

Sustituyendo este resultado en el anterior obtenemos:

$$p_g(s)=p_f(r) * [1/p_f(r)]_{r=T^{-1}(s)}=1_{r=T^{-1}(s)}=1 \text{ para } 0 \leq s \leq 1$$

Por tanto, cuando se considera como función de transformación la acumulación de la función de densidad de probabilidad de intensidades (función de distribución), la imagen resultante ofrece una función densidad de probabilidad uniforme, independiente de la función de densidad de probabilidad original.

Una demostración, también simple, y más en la línea de la teoría de las probabilidades sería la siguiente. Sea F la función de distribución de la variable aleatoria r, esto es:

$$F(kr)=\int p_f(w)dw=p(r \leq kr)$$

La función de distribución de la variable aleatoria $s=F(r)$ es :

$$F(ks)= P(s \leq ks)=P(F(r) \leq ks)$$

Si F es continua y monótona creciente, $F(r) \leq ks$ implica que $r \leq F^{-1}(ks)$ y, por lo tanto la expresión anterior se puede poner de la siguiente manera:

$$F(ks) = P(F(r) \leq ks) = P(r \leq F^{-1}(ks)) = F(F^{-1}(ks)) = ks$$

lo cual sólo puede ser cierto si la variable aleatoria s tiene una función de densidad de probabilidad uniforme en el intervalo $[0,1]$. Por consiguiente, si se toma como función de transformación la función de distribución se obtiene una variable con función de densidad de probabilidad uniforme.

3.1.2 CASO DISCRETO

Para poder aplicar esta idea al procesamiento de imágenes digitales, estos conceptos deben ser formulados de forma discreta.

Sea $P_f(r_k)$ la frecuencia relativa de la intensidad r_k , definida como:

$$P_f(r_k) = n_k / n \text{ con } 0 \leq r_k \leq 1, \quad k=0,1,\dots,L-1$$

donde L es el número de niveles discretos de intensidades (niveles de grises), n_k es el número de veces que el nivel de gris r_k aparece en la imagen y “ n ” el número total de píxeles de la imagen.

Así, $p_f(r_k)$ es una estimación de la probabilidad de que la intensidad de un determinado píxel de la imagen sea r_k . El histograma de la imagen viene dada por la representación de $p_f(r_k)$ frente a r_k .

La función que transforma la imagen en otra con un histograma igualado (“uniforme”) viene dada por:

$$S_k = T(r_k) = \sum_{j=0}^k n_j / n = \sum_{j=0}^k p_f(r_j)$$

Esta técnica recorre la imagen original píxel a píxel transformando el nivel de gris encontrado de acuerdo con la ecuación anterior. Obsérvese como las condiciones impuestas a la función de transformación T permiten que la imagen no se distorsione. De esta manera con la primera condición obligamos a que los nuevos valores de niveles grises asignados estén en todo momento dentro del rango de intensidades permitidos y con la segunda condición se garantiza la asignación de un único valor de intensidad. Por último, la condición de monotonidad creciente impide que los niveles de grises de los píxeles de la imagen puedan invertirse entre ellos, con la consiguiente distorsión cromática.

Estas mismas condiciones permiten que la imagen original pueda recuperarse, teóricamente, mediante la transformación inversa discreta:

$$R_k = T^{-1}(s_k)$$

En la práctica, la aplicación de T hace que se pierdan niveles de grises y, por tanto, información, por lo que la transformación inversa no podrá recuperar la misma imagen original.

3.2 TRANSFORMACIÓN DE LA ESCALA DE GRISES

La idea que subyace en este tipo de procesamiento no es más que la de aprovechar todos los niveles de la escala de grises.

Si nos fijamos en la figura 1.9 donde se muestra un diagrama de intensidades se observa cómo existe una amplia banda de niveles de grises que no son utilizados en la imagen. Aprovechando estos niveles, la transformación de la escala de grises permite dilatar el rango de intensidades con el consiguiente aumento de contraste de la imagen.

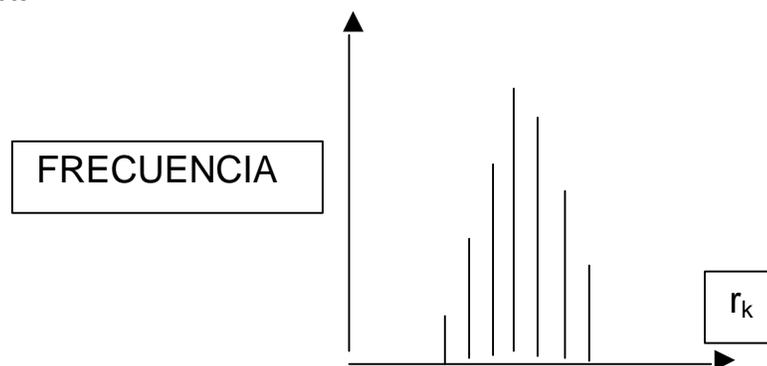


Figura 1.9.-*Histograma de una imagen en la cual no todos los niveles de grises son utilizados.*

En este caso la información proporcionada por el histograma de la imagen a procesar no interviene directamente en la obtención de la función de transformación, aunque sí es tomada en cuenta.

La modificación de la escala de grises de una imagen f con niveles de intensidades discretos r_k viene dada por una transformación:

$$S_k = T(r_k) \quad k=0,1,\dots,L-1$$

donde S_k son los niveles de grises (también discretos) de la imagen resultante, y L es el número de estos niveles.

Se supone que ambas escalas disponen del mismo rango de niveles, es decir, si r_k toma valores entre 0 y 255, también los s_k se encuentran comprendidos en dicho rango. Esta suposición es usual para la gran mayoría de aplicaciones en procesamiento de imágenes.

Considérese el aumento de contraste de una imagen con histograma como el mostrado en la **figura 1.9**, en la cual sus intensidades no ocupan por completo el rango disponible. Suponiendo que para una imagen dada f los valores de todos los píxeles de la misma se encuentran comprendidos en el rango:

$$a \leq r_k = f(i, j) \leq b \quad \forall \text{ pixel } (i, j) \text{ perteneciente a la imagen } f \text{ donde } [a, b] \text{ es un subintervalo del rango de intensidades } [r_0, r_{L-1}].$$

La transformación lineal:

$$S_k = [(r_{L-1} - r_o) / (b - a)] * [r_k - a] + r_o \quad a \leq r_k \leq b$$

“ estira “ y traslada la escala de grises para ocupar el rango completo $[r_o, r_{L-1}]$. Para ver este efecto basta con ver la figura 1.10.

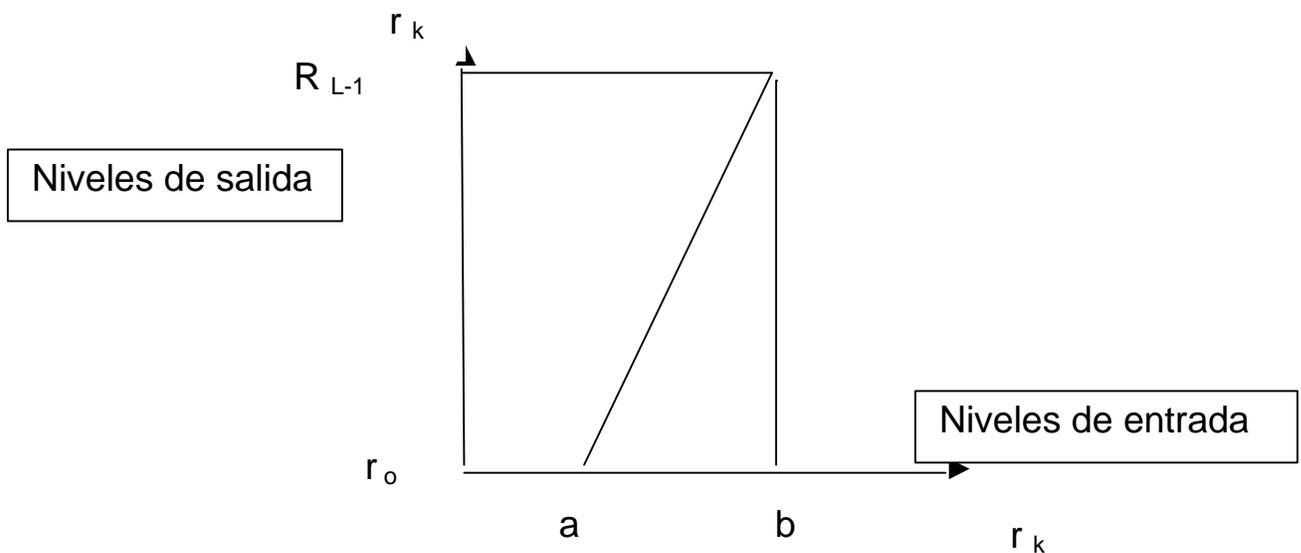


Figura 1.10.- Dilatación del intervalo $[a, b]$ al rango completo de intensidades $[r_o, r_{L-1}]$.

Esta transformación también puede utilizarse cuando la mayoría de los niveles de grises de la imagen dada quedan dentro del subrango $[a, b]$. En este caso puede utilizarse la función:

$$S_k = \begin{cases} [(r_{L-1}-r_0)/(b-a)]*(r_k-a)+ r_0 & a \leq r_k \leq b \\ r_0 & r_k < a \\ r_{L-1} & r_k > b \end{cases}$$

Esta transformación dilata el intervalo $[a,b]$ de la escala original a la vez que comprime los intervalos $[r_0, a]$ y $[b, r_{L-1}]$ a un solo punto. Esta compresión puede tolerarse si son pocos los puntos en estos dos últimos intervalos, esto es, si es poca la información perdida.

De forma general, se puede dilatar determinadas regiones a costa de la compresión de otras, siempre y cuando, el objetivo prioritario sea el de conseguir aumento de contraste en bandas concretas, y no importe demasiado la pérdida de información en las regiones comprimidas.

Finalmente, señalar que las transformaciones de niveles de grises hasta ahora comentadas son lineales a trozos, sin embargo, pueden implantarse otro tipo de transformación matemática no-lineales como la cuadrática, logarítmica, etc., con la única restricción de que los niveles de grises obtenidos queden dentro del rango permitido, esto es:

$$r_o \leq T(r_k) \leq r_{L-1} \quad \forall r_k \text{ con } r_o \leq r_k \leq r_{L-1}$$

3.3 REALCE LOCAL

En determinadas circunstancias suele ser conveniente plantear el realce a partir de la distribución de intensidades sobre el entorno de vecindad de cada píxel de la imagen. El procedimiento general para la aplicación de un realce local consiste en definir una ventana $n \times m$ que es posicionada píxel a píxel por la imagen, operando solamente en dicho entorno en base a atributos de las intensidades de sus píxeles.

Dos de los atributos que más se utilizan son la media y la varianza. La media de intensidades nos da una medida del brillo medio, mientras que la varianza proporciona una estimación del contraste de la región considerada de la imagen. Así pues, cuanto mayor sea la media, mayor será el brillo medio, y mientras mayor sea la varianza, mayor será el contraste.

A continuación se muestra una transformación típica basada en estos conceptos, donde los valores de intensidad de la imagen de entrada $f(i,j)$ son modificados píxel a píxel obteniéndose una nueva imagen $g(i,j)$ de la siguiente forma:

$$g(i,j) = A(i,j) * [f(i,j) - m(i,j)] + m(i,j) \quad (1.1)$$

con:

$$A(i,j)=K*M/\sigma(i,j)$$

donde $m(i,j)$ y $\sigma(i,j)$ son la media y la desviación típica calculadas sobre un entorno de vecindad centrado en el píxel (i,j) , M es la media global de $f(i,j)$, K es una constante que toma valores entre 0 y 1.

La utilización del término $A(i,j)$ permite amplificar las variaciones locales según una ganancia local. De esta forma, en zonas de bajo contraste $\sigma(i,j)$ será pequeña y, por tanto, la ganancia $A(i,j)$ grande, produciéndose valores elevados de $g(i,j)$.

En la función de transformación 1.1 se ha sumado la media $m(i,j)$ para de esta forma restablecer el brillo medio de intensidad. Para evitar saturaciones en la intensidad resultante $g(i,j)$, la ganancia A suele limitarse dentro de un intervalo $[A_{min}, A_{max}]$.

3.4 SUAVIZADO

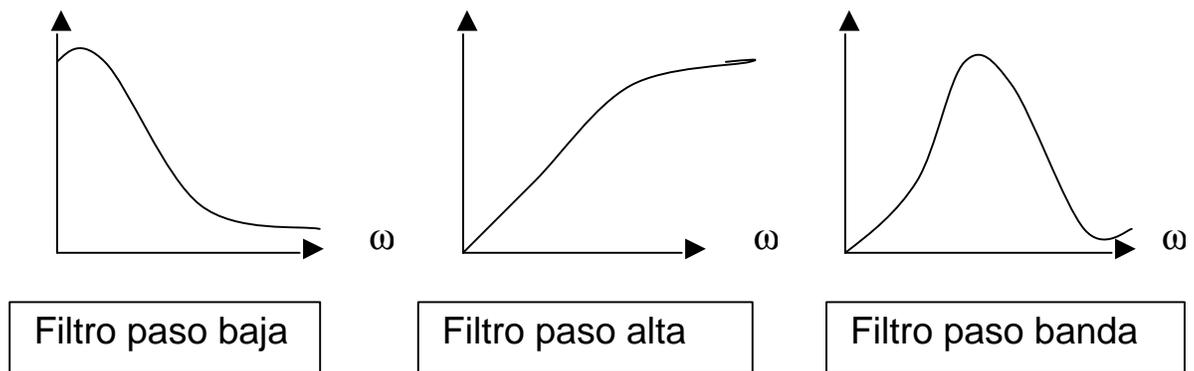
Las operaciones de suavizado tienen por objeto reducir ruido y/o efectos espurios que pueden presentarse en una imagen a consecuencia del proceso de captura, digitalización y transmisión. Su aplicación es normalmente necesaria antes de la aplicación de un detector de bordes.

De acuerdo con la teoría de filtrado de señales, la reducción de ruido se puede realizar en el dominio de las frecuencias mediante la aplicación de un filtro paso-baja. Como es sabido, la multiplicación de

dos señales en el dominio de las frecuencias equivale a la convolución de éstas en el dominio espacial, es decir:

$$G(W)=F(W)H(W) \Leftrightarrow g(n)=f(n) \otimes h(n)$$

donde $G(W)$, $F(W)$, $H(W)$ son las transformadas de Fourier de las señales unidimensionales $g(n)$, $f(n)$ y $h(n)$ respectivamente. En el caso de imágenes digitales estas funciones serán, lógicamente, bidimensionales y discretas. Algunas funciones típicas empleadas en el filtrado de señales son:



Las filtros arriba dibujados están representados en el dominio de las frecuencias y sus homólogos en el dominio del espacial son:

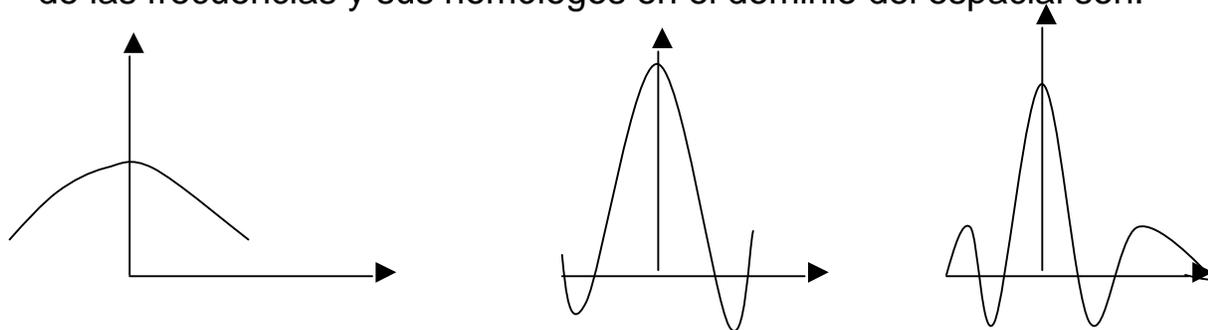


Figura 1.11 *Respuestas impulsionales correspondientes (dominio espacial).*

Aunque en el campo del procesamiento de imágenes es habitual trabajar en frecuencias, en aplicaciones de visión por computador estos filtrados suelen hacerse directamente en el dominio espacial, ya que el coste computacional de la transformada rápida de Fourier (FFT) suele resultar excesivo.

En lo que sigue presentaré las técnicas más comúnmente empleadas para suavizar una imagen. En particular, se estudian el promediado del entorno de vecindad, filtrado gaussiano, promediado de imágenes y, finalmente, se presenta el filtrado de mediana.

3.4.1 PROMEDIADO DEL ENTORNO

Bajo el nombre de promediado del entorno se agrupan una serie de técnicas que tienen en cuenta el nivel de gris de los píxeles vecinos sobre el que se pretende actuar. Sobre esta base se desarrollan una serie de variantes que difieren, principalmente, en la forma de escoger a los píxeles vecinos y en la ponderación que se le da a cada uno de ellos. En cualquier caso, esta operación se lleva a cabo mediante la convolución de la imagen con una ventana o máscara.

El principal inconveniente de estas técnicas es el enturbiamiento que se produce en la imagen, provocando el difuminado de los bordes, por lo que deben ser usados con cierta precaución.

Los distintos filtros de promediado del entorno pueden clasificarse en tres tipos:

- Los filtros paso-baja, que pueden eliminar ruido caracterizado por un determinado contenido de altas frecuencias. La respuesta impulsional de estos filtros es infinita (IIR: Infinite Impulse Response) y, por tanto, actúan sobre una extensión infinita de la imagen presentando serios inconvenientes de implementación. Un ejemplo de ello sería un filtro paso-baja ideal.
- Filtros de respuesta impulsional finita (FIR: Finite Impulse Response), con actuación sobre una región finita de la imagen. Aunque interesantes desde el punto de vista computacional, presenta el inconveniente de actuar sobre una extensión infinita de frecuencias. Un ejemplo es el filtro de la media simple que comentaré a continuación.
- Filtros que pueden ser considerados como una combinación de los dos anteriores, puesto que optan por un compromiso entre la actuación en el dominio de frecuencia y espacial. Este compromiso viene precisado por un parámetro de incertidumbre ΔU (Marr y Hildreth 1980).

$$\Delta U = \Omega X \quad (1.2)$$

donde Ω y X son la extensión de actuación en el dominio de frecuencias y espacial, respectivamente.

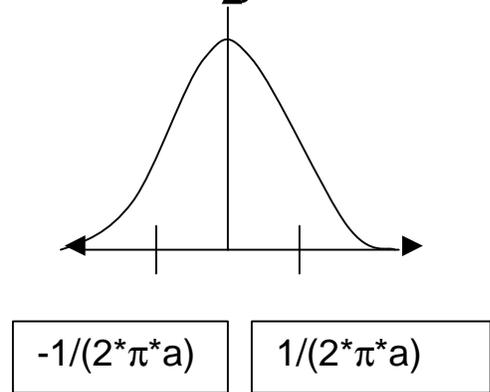
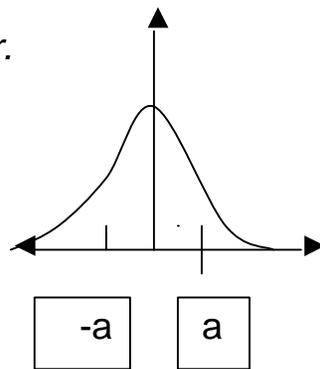
Lo realmente relevante de esto, es que la incertidumbre ΔU está acotada inferiormente, es decir:

$$\Delta U \geq 1/2$$

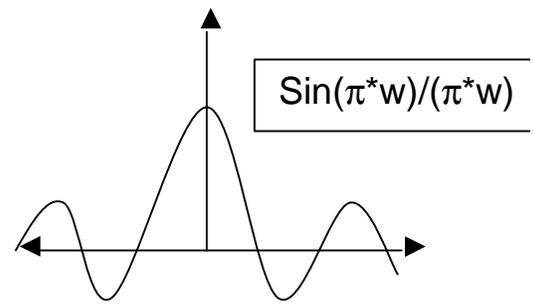
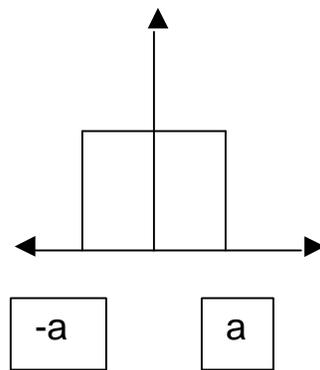
Un operador de suavizado es, por tanto, mejor cuanto más próximo a 1/2 sea su parámetro ΔU .

Figura 1.12.- Transformada de Fourier de una función gaussiana y de un pulso rectangular.

Gaussiana



Pulso rectangular



3.4.1.1 MEDIA SIMPLE

Dada una imagen $f(i,j)$, el procedimiento consiste en generar una nueva imagen $g(i,j)$ cuya intensidad para cada píxel se obtiene promediando los valores de intensidad de los píxeles de $f(i,j)$ incluidos en un entorno de vecindad predefinido, es decir, la imagen suavizada se obtiene como:

$$g(i,j) = (1/p) \sum f(m,n)$$

donde el sumatorio se extiende a un subconjunto de píxeles situados en el entorno de (i,j) incluyendo (i,j) .

Si se emplea un entorno 3x3, el proceso es equivalente a convolucionar la imagen con una máscara de la forma:

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

El tamaño de la máscara es importante. Cuanto mayor sea ésta se consigue una mayor difuminación de los bordes. Así hemos podido comprobar como con una máscara de 9x9 se produce una mayor difuminación de los bordes que con una 3x3.

La imagen que se obtiene al aplicar un filtro 3x3 de media simple sobre la quinta figura de la imagen Gatlin es la siguiente:



No obstante el efecto de enturbiamiento de los bordes puede ser reducido dejando sin cambiar las regiones de la imagen con variación grande del nivel de gris (previsiblemente bordes), es decir:

$$g(x,y) = \begin{cases} (1/p) * \sum f(m,n) & \text{si } |f(x,y) - (1/p)\sum f(m,n)| < T \\ f(x,y) & \text{en otro caso} \end{cases}$$

donde T es un umbral positivo. El problema fundamental que presenta esta variante es que no actúa en los puntos espurios.

3.4.1.2 PROMEDIADO SELECTIVO Y PROMEDIADO PONDERADO

Si fuera posible realizar el promediado teniendo en cuenta sólo determinados píxeles vecinos, se podría evitar el difuminado de características de interés como bordes, esquinas, líneas, y puntos de interés. Esta técnica, conocida como promediado selectivo, se puede abordar mediante un gran número de variantes.

Un posible enfoque consiste en añadir un detector de bordes y promediar sólo en los píxeles donde éstos no hayan sido detectados. Otra idea más refinada es realizar un promediado direccional, es decir, si el borde es detectado se promedia sólo con los píxeles que se encuentran en la dirección del borde. El principal inconveniente de este tipo de promediado es el coste computacional que resulta de añadir el detector de bordes, lo que lo hace inviable en la mayoría de las aplicaciones.

Otra alternativa consiste en decidir qué píxeles vecinos entran en el promediado, en base al parecido con el píxel que se está procesando. Como ejemplo de esta técnica, considérense los siguiente vecinos del píxel M a procesar:

A B C D E
F G H I J
K L M N O
P Q R S T
U V W X Y

de donde se toman las siguientes submáscaras:

A B C C D E G H I K L M M N O
F G H H I J L M N P Q R R S T
K L M M N O Q R S U V W W X Y

Para cada submáscara se mide la variación del nivel de gris y se reemplaza el píxel M con el promedio de la submáscara cuyos píxeles sean más parecidos entre ellos.

Por otro lado el objetivo de realizar un promedio ponderado, es decir, con distintos pesos para cada píxel, es igualmente evitar el difuminado de características de interés de la imagen. La idea y los

procedimientos son similares a los comentados para el promediado selectivo.

La forma de promediado ponderado más empleada es el filtro de Gauss, que por su importancia se tratará en el siguiente apartado.

3.4.2 FILTRO DE GAUSS

Una distribución gaussiana con desviación típica σ y media μ viene dada por:

$$g_{\sigma}(x) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$$

La convolución de la función $g_{\sigma}(x)$ con una señal $f(x)$ da lugar a una nueva señal suavizada $h(x)$, donde el valor en cada punto es el resultado de promediar con distintos pesos los valores vecinos a ambos lados de dicho punto. En este suavizado, la desviación típica σ juega un papel importante a la hora de controlar el grado de suavizado de este operador. Cuanto mayor sea σ , más se tienen en cuenta los puntos lejanos, y, por tanto, se realizará un mayor suavizado. Conforme σ disminuye, los valores de los puntos vecinos van

recibiendo menor ponderación, con lo que la reducción de ruido será menor.

Si extendemos la función anterior a dos dimensiones como se trabaja en imagen tendremos:

$$g_{\sigma}(x,y)=(1/(2\pi\sigma^2))*\exp[(-1/2)(x^2+y^2)/\sigma^2]$$

0	0	1	2	3	4	4	5	4	4	3	2	1	0	0
0	1	2	4	6	8	10	11	10	8	6	4	2	1	0
1	2	4	8	12	16	20	21	20	16	12	8	4	2	1
2	4	8	14	21	29	35	37	35	29	21	14	8	4	2
3	6	12	21	32	44	54	57	54	44	32	21	12	6	3
4	8	16	29	44	61	73	78	73	61	44	29	16	8	4
4	10	20	35	54	73	88	94	88	73	54	35	20	10	4
5	11	21	37	57	78	94	100	94	78	57	37	21	11	5
4	10	20	35	54	73	88	94	88	73	54	35	20	10	4
4	8	16	29	44	61	73	78	73	61	44	29	16	8	4
3	6	12	21	32	44	54	57	54	44	32	21	12	6	3
2	4	8	14	21	29	35	37	35	29	21	14	8	4	2
1	2	4	8	12	16	20	21	20	16	12	8	4	2	1
0	1	2	4	6	8	10	11	10	8	6	4	2	1	0
0	0	1	2	3	4	4	5	4	4	3	2	1	0	0

Figura 1.13 *Máscara de convolución 15x15*

En el caso discreto, la convolución de una imagen $f(i,j)$ con el equivalente discreto de la anterior expresión viene dado por:

$$F(i,j) = \frac{1}{(2\pi\sigma^2)} \sum_{k,l} f(i+k, j+l) \exp\left[-\frac{1}{2\sigma^2}(k^2+l^2)\right]$$

KI

Aunque este tipo de suavizado también tiene el problema de difuminado de los bordes, éste no es tan acusado como en el caso de la media simple.

En la práctica, los sumatorios se limitan a un determinado entorno de vecindad del píxel analizado, lo que se traduce en la convolución de la imagen con una máscara cuadrada cuyo tamaño dependerá de la desviación típica σ escogida. A medida que aumenta el valor de σ , los términos de la función gaussiana alejados de la media (0 en este caso) son mayores y no deben, por tanto, ser despreciados.

Se han propuesto diferentes relaciones entre σ y el tamaño W del operador. Una de las más comúnmente aceptadas es:

$$W \geq 3c$$

donde $c = 2\sqrt{2}\sigma$ es el tamaño del lóbulo central de la gaussiana.

Como puede apreciarse, en el caso discreto, un operador $g(i,j)$ se dice que es separable si puede ponerse como:

$$g_{\sigma}(x,y)=g_1(x)g_2(y)=1/(\sigma\sqrt{2\pi})\exp[(-1/2)(x/\sigma)^2](1/(\sigma\sqrt{2\pi})\exp((-1/2)(y/\sigma)^2))$$

De forma equivalente, en el caso discreto se dice que un operador $g(i,j)$ es separable si puede ponerse como:

$$g_{\sigma}(i,j)=g_1(i)g_2(j)$$

siendo g_1 un vector de filas y g_2 un vector de columnas. De esta manera, el filtro gaussiano se aplica en dos etapas: En la primera se realiza una convolución por filas y al resultado obtenido se le aplica una convolución por columnas. Así se consigue reducir el número de operaciones de N^2 a $2N$.

De acuerdo con los trabajos de Marr y Hildreth, la función real que minimiza el parámetro de incertidumbre dado por la ecuación 1.2 es la función gaussiana. Esta circunstancia hace que el filtro gaussiano sea considerado como un operador óptimo de suavizado, en especial, para la eliminación de ruido previa a la detección de bordes de la imagen.

3.4.3 PROMEDIADO DE IMÁGENES

En esta técnica se promedian píxeles de distintas imágenes de una misma escena cuya única diferencia teórica es el ruido.

Sea $g(x,y)$ una imagen real con ruido compuesta por la suma de una imagen ideal $f(x,y)$ y de un ruido $\eta(x,y)$:

$$g(x,y) = f(x,y) + \eta(x,y)$$

donde se supone que $\eta(x,y)$ es un ruido blanco, es decir, de media cero y matriz de covarianzas diagonal (ruido no correlado).

Si $\bar{g}(x,y)$ es el promedio de M imágenes ruidosas:

$$\bar{g}(x,y) = (1/M) \sum g_i(x,y)$$

se verifica que:

$$E[\bar{g}(x,y)] = f(x,y) \quad (1.3)$$

y

$$\Sigma_{\bar{g}}(x,y) = (1/M) \Sigma_{\eta}(x,y) \quad (1.4)$$

De acuerdo con las expresiones 1.3 y 1.4, la imagen promedio de M imágenes ruidosas tiende a reducir el ruido inherente a las imágenes reales: Lógicamente, cuanto mayor es M más ruido se elimina.

Para que esta técnica pueda aplicarse es imprescindible que la escena no varíe, es decir, que no varíen ni los objetos ni las condiciones de iluminación.

3.4.4 FILTRADO DE MEDIANA

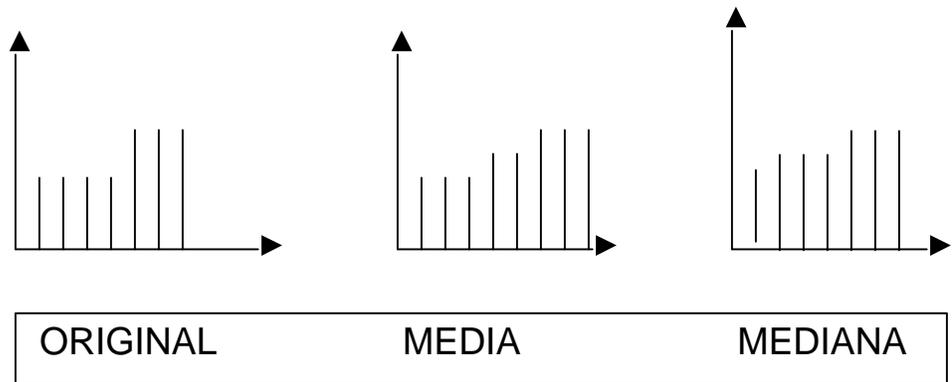
Una de las principales dificultades del promediado del entorno es que desdibuja los contornos y otros detalles de la imagen. Una alternativa es el uso de filtros de mediana, que consiste en reemplazar el nivel de gris de cada píxel por la mediana de los niveles de grises de los vecinos. A diferencia de los métodos vistos hasta ahora, esta técnica no se implementa en base a la convolución de la imagen con una determinada máscara (método lineal), sino que es un método no-lineal.

Sea $\{a_1, a_2, \dots, a_N\}$ una secuencia discreta de un número impar de valores ordenados creciente o decrecientemente. Se define la mediana de esta secuencia como el elemento $a_{(N-1)/2}$. Por ejemplo, se consideran los 4-vecinos de “ p ” cuyas intensidades son 1,2,9,4 y 5, el píxel “ p ” se reemplazará por el valor 4, ya que éste es el valor central en la secuencia ordenada 1,2,4,5 y 9.

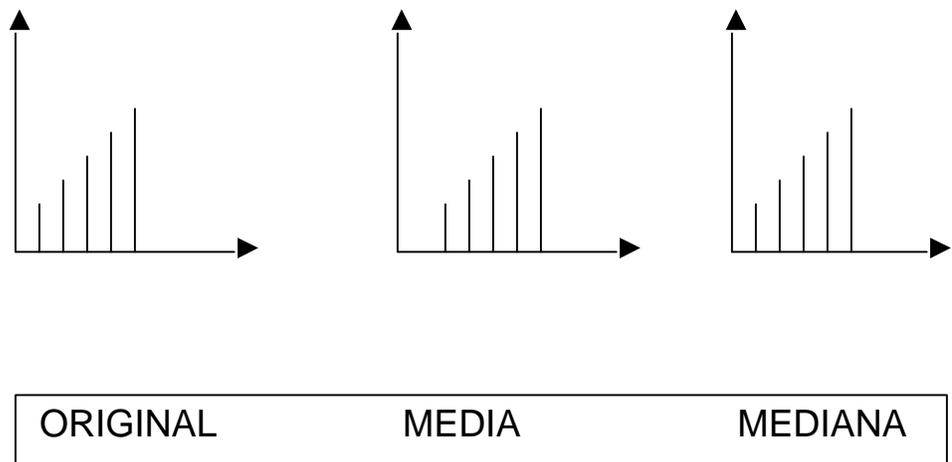
Si comparamos la actuación de un filtro de mediana y de un filtro de media para varias funciones discretas unidimensionales (escalón,

rampa, pulso y triángulo), empleando una ventana de tamaño 5 apreciamos los siguientes resultados:

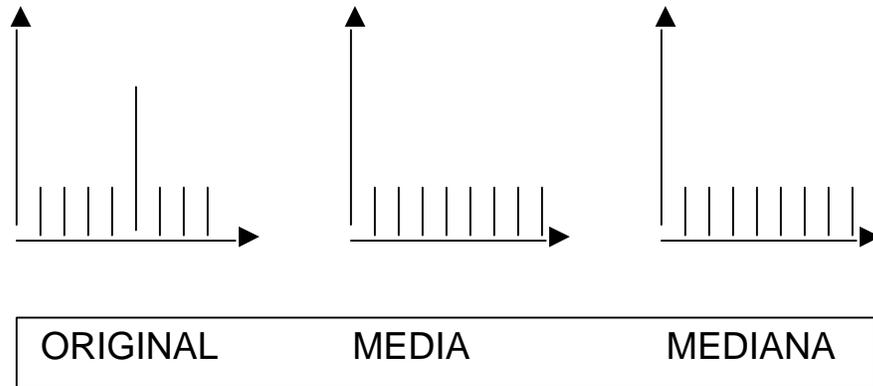
- 1) Sobre una función escalón observamos que en el filtro de media se produce una distorsión mientras que en el de mediana no.



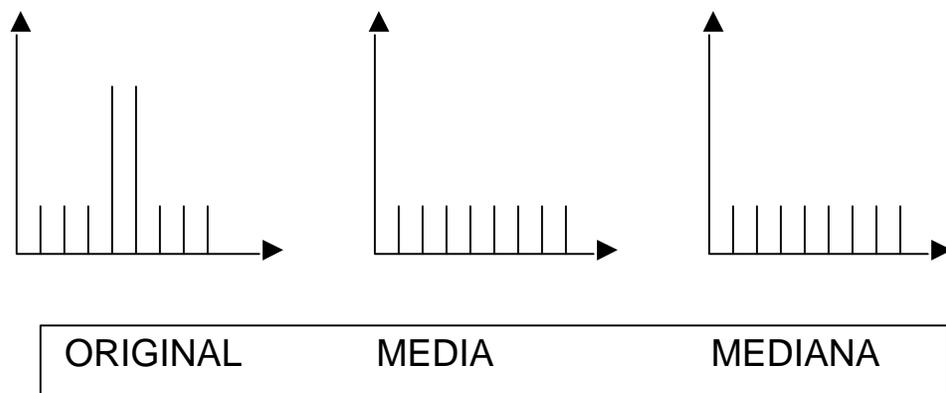
- 2) Sobre la función rampa ni la función de media ni la mediana provocan ninguna distorsión en la imagen resultante.



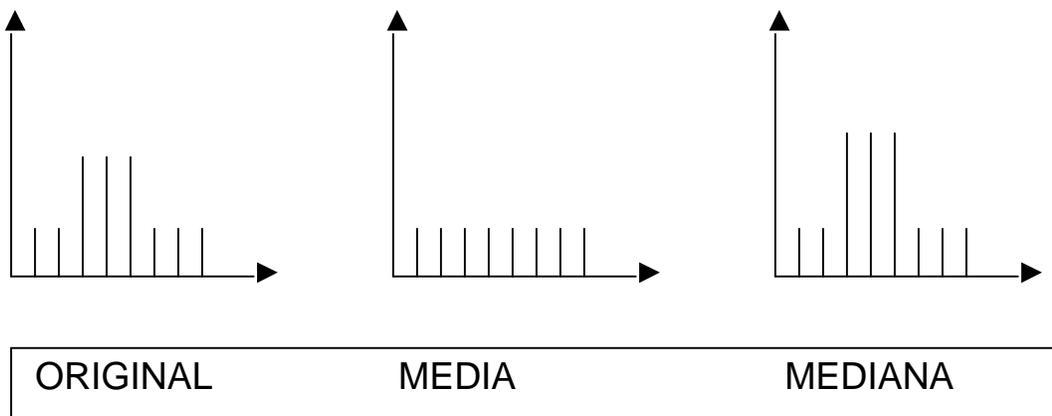
- 3) Sobre un pulso simple tanto la media como la mediana produce un recorte del pulso distorsionando la señal.



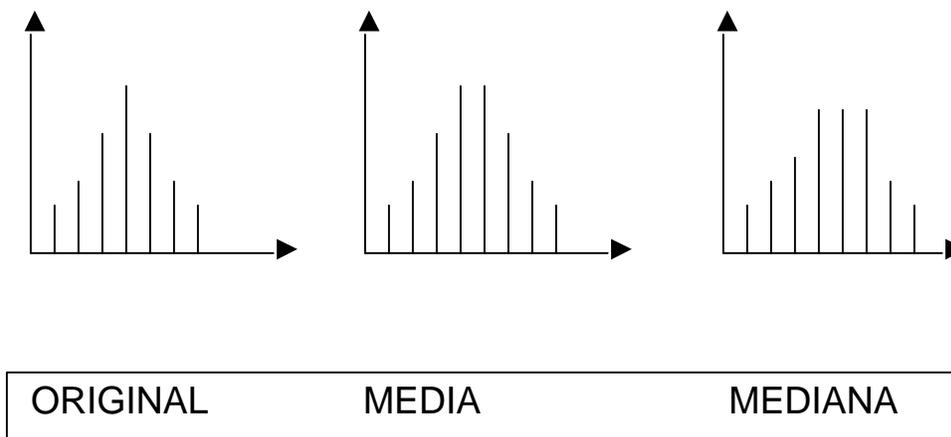
- 4) Sobre un pulso doble se produce el mismo efecto del punto 3.



- 5) Sobre un pulso triple se produce una distorsión de la media pero no de la mediana.



- 6) Sobre una entrada triangular se produce una distorsión tanto en la media como en la mediana.



Puede afirmarse que, el comportamiento del filtro de mediana es mejor que el del promediado del entorno para eliminar efectos espurios y para preservar los bordes de la imagen. Presenta, sin embargo, el inconveniente de que pierde los detalles finos como líneas delgadas, puntos aislados y redondea las esquinas.

Otro efecto que puede producir el filtro de mediana es el desplazamiento de los bordes. Dicho problema debe ser tenido en cuenta en aplicaciones donde se requiera una gran precisión en la localización de éstos (Davies, 1990).

A la hora de implementar un filtro de mediana el procedimiento básico sería:

- Ordenar los $2N+1$ valores (intensidades) del entorno de vecindad de cada píxel.
- Varias posibilidades:
 - Tomar como intensidad del píxel la mediana de los valores anteriores.
 - Eliminar los k mayores y los k menores y promediar con el resto.
 - Cualquier otra forma.

Hay que darse cuenta que no es necesario volver a ordenar todos los valores de la ventana cada vez que ésta se desplaza a un nuevo píxel, sino que ésta se va reciclando, es decir, se eliminan de la secuencia ordenada los píxeles que se quedan fuera y se insertan los nuevos.

A pesar de tener en cuenta implementaciones como ésta, el principal inconveniente del filtro de mediana es su elevado coste

computacional. Para acelerarlo se han propuesto diversos algoritmos, entre los que se encuentra, por ejemplo, el método de Ataman (Davis, 1990). La idea básica de este método radica en que sólo el bit más significativo (MSB) de los píxeles que pertenecen a la ventana analizada es relevante a la hora de calcular el MSB de la mediana de los niveles de grises de esa ventana. Esto sugiere la posibilidad de determinar la mediana en un número de operaciones del orden del número de bits que se usan para las intensidades de los píxeles, que generalmente son ocho.

Otro algoritmo de cálculo rápido, llamado filtro de la pseudo-mediana, consiste en hallar mínimos (o máximos) de subsecuencias y posteriormente hallar el máximo (o mínimo) de los mínimos (o máximos) obtenidos en el paso anterior. Este algoritmo, aunque de fácil implementación, presenta el inconveniente de que no suele dar la mediana verdadera, de ahí el prefijo “pseudo” .

4. DETECCIÓN DE BORDES.

4.1 INTRODUCCIÓN A LA DETECCIÓN DE BORDES

Los bordes de una imagen se pueden definir como transiciones entre dos regiones de niveles de gris significativamente distintos. Éstos suministran una valiosa información sobre las fronteras de los objetos que puede ser utilizada para la segmentación de la imagen, reconocimiento de objetos, visión estéreo, etc.

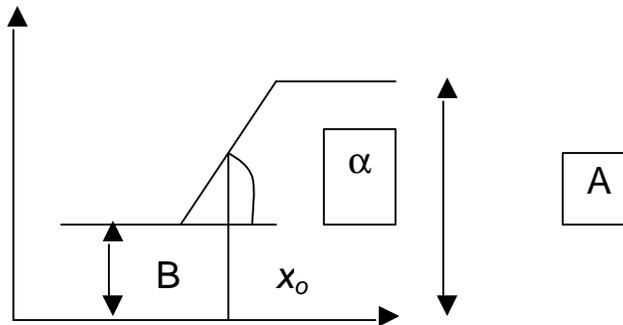


Figura 1.14.- Modelo unidimensional y continuo de un borde ideal.

En la **figura 1.14** se muestra un modelo unidimensional y continuo de un borde. Este modelo representa una rampa desde un nivel de gris bajo “B” a uno alto “A”, con los siguientes parámetros:

- Variación de intensidad $H = A - B$
- Ángulo de inclinación de la rampa “ α ”
- Coordenada horizontal x_0 donde se encuentra el punto medio de la rampa.

Un operador que proporcionase los valores de x_0 y H daría unos datos valiosísimos sobre la imagen, ya que proporcionaría la amplitud del borde, y lo localizaría con exactitud dentro de la imagen.

En las imágenes reales los bordes nunca se ajustan totalmente al modelo anterior. Las causas de ello son diversas, destacándose las siguientes:

- Las imágenes son discretas.
- Están afectadas por un ruido diverso.
- El origen de los bordes puede ser muy diverso: bordes de oclusión, superficies de diferente orientación, distintas propiedades reflectantes, distinta textura, efectos de iluminación (sombras y/o reflejos) etc.

Por todo esto, debido a la complejidad a la hora de detectar los bordes de una imagen, se consideran tres tipos de errores.

- Error de detección. Un operador es un buen detector si la probabilidad de detectar el borde es alta cuando éste realmente existe en la imagen, y baja cuando éste no existe.
- Error en la localización. Un operador localiza bien un borde cuando la posición que proporciona coincide con la posición real del borde en la imagen.

Ambos errores están estrechamente ligados a los problemas anteriormente señalados, y muy especialmente a la presencia de ruido de la imagen. En la práctica, la calidad en la detección y en la localización están en conflicto, como se verá más adelante en este capítulo.

- Respuesta múltiple. Varios píxeles son detectados en un único borde.

A continuación vamos a poner tres ejemplos que ilustren los errores anteriores.

0	1	0	0
0	1	0	0
0	1	0	0
0	1	0	0

Borde real

0	0	1	0
1	0	0	0
0	0	1	0
1	0	0	0

Pobre detección

0	1	0	0
0	0	1	0
0	0	0	1
0	0	1	0

Pobre localización

0	1	0	0
0	1	1	0
1	1	1	1
1	1	0	0

Múltiple respuesta

La mayoría de las técnicas para detectar bordes emplean operadores locales basados en distintas aproximaciones discretas de la primera y segunda derivada de los niveles de grises de la imagen, si bien existen otras posibilidades para ello, como el empleo de patrones de bordes ideales. A continuación se recogen las técnicas más significativas de cada uno de estos enfoques.

4.2 OPERADORES BASADOS EN MÁSCARAS DIRECCIONABLES

Puesto que un borde ideal es esencialmente un patrón con forma de escalón, un método directo de detectar los bordes consiste en correlar la imagen con patrones en toda las orientaciones posibles. En la practica, el número de orientaciones se limita a ocho (giradas 45 grados). En la **figura 1.15** se muestra, a modo de ejemplo, las máscaras patrones propuestas por Kirsch (1971).

El valor del gradiente en cada píxel de la imagen se obtiene como el máximo de las respuestas a cada una de las máscaras, mientras que la dirección será la indicada por la orientación de dicha máscara.

Aunque se han propuesto otras muchas máscaras direccionales, el alto coste computacional y la escasa precisión en la determinación del gradiente desaconsejan su utilización en la mayoría de las aplicaciones.

5	-3	-3
5	0	-3
5	-3	-3

ESTE

-3	-3	-3
5	0	-3
5	5	-3

N-E

-3	-3	-3
-3	0	-3
5	5	5

N

-3	-3	-3
-3	0	5
-3	5	5

N-W

-3	-3	5
-3	0	5
-3	-3	5

W

-3	5	5
-3	0	5
-3	-3	-3

S-W

5	5	5
-3	0	-3
-3	-3	-3

S

5	5	-3
5	0	-3
-3	-3	-3

S-E

Figura 1.15.- Máscaras orientadas de Kirsch para la detección de bordes.

4.3 OPERADORES BASADOS EN LA PRIMERA DERIVADA

Como se sabe, la derivada de una señal continua proporciona las variaciones locales con respecto a la variable, de tal forma que el valor de la derivada es mayor cuanto más rápidas son estas variaciones.

Si consideramos una señal continua $f(x)$ que representa un borde en una dimensión, parece lógico considerar que la localización exacta de este borde venga dada por el punto de inflexión de $f(x)$, es decir, por el máximo de la función $f'(x)$ (punto x_0). De esta manera la derivada de $f(x)$ permite determinar “ x_0 ” de una manera fácil, al tiempo que proporciona una estimación sobre la magnitud y el sentido de la variación (creciente si $f'(x)$ es positivo y, decreciente, en sentido contrario).

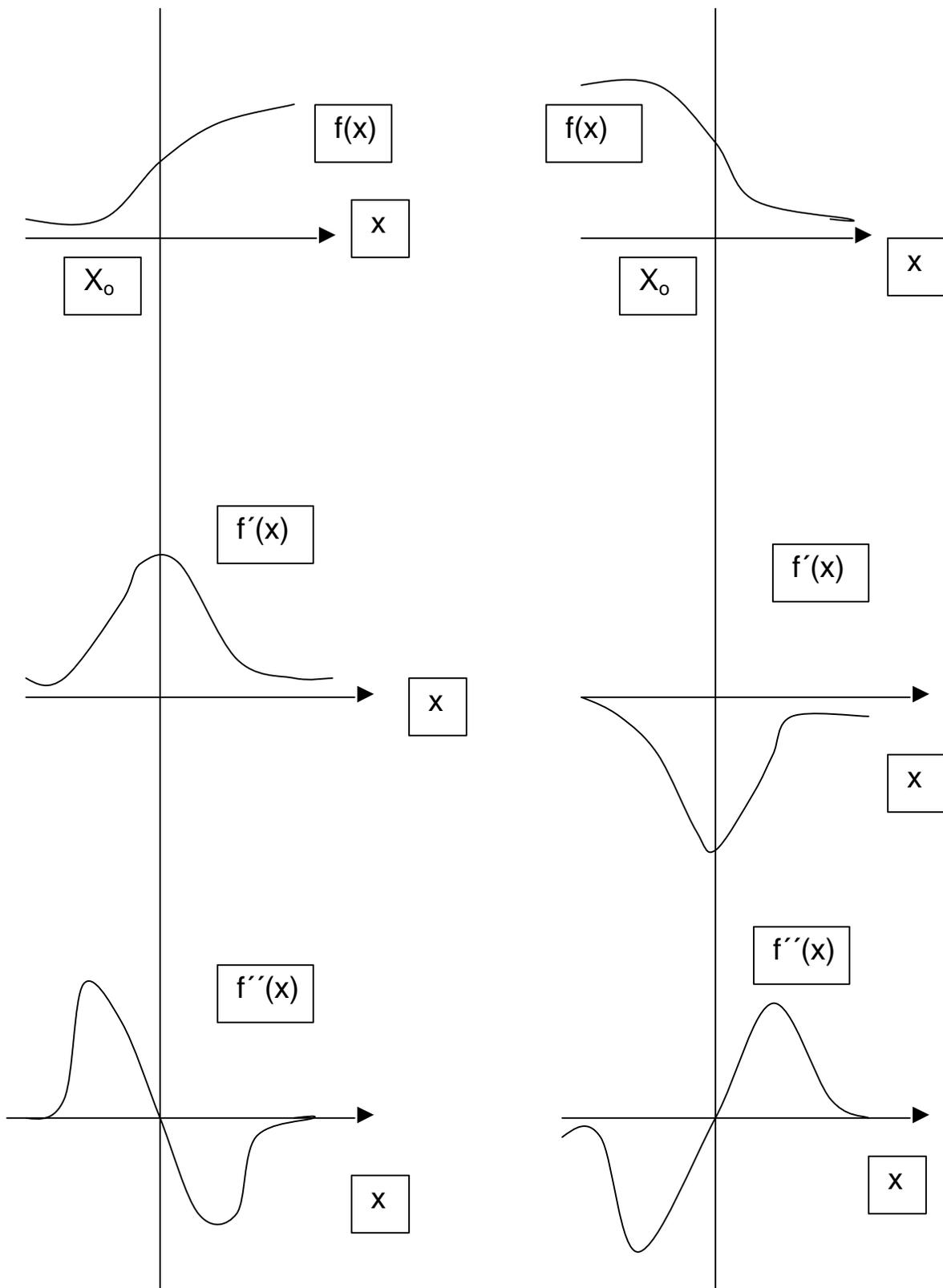


Figura 1.16.- Representación típica de un borde de una dimensión. Primeras y segundas derivadas.

En el caso de funciones bidimensionales $f(x, y)$, la derivada es un vector que apunta en la dirección de máxima variación de $f(x, y)$ y cuyo módulo es proporcional a dicha variación. Este vector, denotado por $\nabla f(x, y)$, se denomina gradiente y se define como:

$$\nabla f(x, y) = \begin{pmatrix} \partial f(x, y) / \partial x \\ \partial f(x, y) / \partial y \end{pmatrix} = \begin{pmatrix} f_x(x, y) \\ f_y(x, y) \end{pmatrix}$$

Hay que hacer notar que el gradiente en un punto (x, y) viene dado por las derivadas de $f(x, y)$ a lo largo de los ejes coordenados ortogonales “x” e “y”. El módulo y la dirección del gradiente vienen dados por:

$$|\nabla f(x, y)| = \sqrt{[f_x(x, y)]^2 + [f_y(x, y)]^2} \quad (1.5)$$

$$\alpha (x , y) = \text{atan} (f_y(x , y)/f_x (x , y)) \quad (1.6)$$

4.3.1 APROXIMACIONES DISCRETAS DEL OPERADOR GRADIENTE

En el caso bidimensional discreto, las distintas aproximaciones del operador gradiente se basan en diferencias entre los niveles de grises de la imagen. Así, por ejemplo, la derivada parcial $f_x (x , y)$ (gradiente de fila) puede aproximarse por la diferencia de píxeles adyacentes de la misma fila, es decir:

$$f_x(x , y) \approx G_F (i,j) = [F (i , j) - F (i , j-1)]/ T \quad (1.7)$$

o también por la diferencia de píxeles separados:

$$f_x(x , y) \approx G_F (i , j) = [F (i , j+1) - F (i , j-1)]/ 2T \quad (1.8)$$

donde he supuesto que el índice de filas “ i “ crece de arriba abajo, y el de columnas “ j “ de izquierda a derecha. Los factores de escala $1/ T$ y $1/ 2T$ pueden ser omitidos.

El gradiente de fila G_F y de columnas G_C en cada punto se obtienen mediante la convolución de la imagen con sendas máscaras H_F y H_C , esto es:

$$G_F(i, j) = F(i, j) \otimes H_F(i, j)$$

$$G_C(i, j) = F(i, j) \otimes H_C(i, j)$$

donde $H_F(i, j)$ y $H_C(i, j)$ son las respuestas impulsionales del gradiente de fila y columna, respectivamente.

A continuación veremos las máscaras más comunes para obtener el gradiente de una imagen.

OPERADOR GRADIENTE DE FILA GRADIENTE COLUMNA

Diferencia de píxels

0	0	0
0	1	-1
0	0	0

0	-1	0
0	1	0
0	0	0

Diferencia de píxeles separados

0	0	0
1	0	-1
0	0	0

0	-1	0
0	0	0
0	1	0

Para aplicar el operador de Roberts utilizamos las siguientes máscaras:

GRADIENTE FILA

0	0	0
0	0	1
0	-1	0

GRADIENTE COLUMNA

-1	0	0
0	1	0
0	0	0

La aplicación del operador de Prewitt requiere de a utilización de las siguientes máscaras:

GRADIENTE FILA

1	0	-1
1	0	-1
1	0	-1

GRADIENTE COLUMNA

-1	-1	-1
0	0	0
1	1	1

No obstante hay que decir que el filtro de Prewitt se acompaña de un factor de escala $1/3$.

Si en vez de usar los filtros anteriores optamos por un filtro de Sobel las máscaras a emplear serán las siguientes:

GRADIENTE FILA

1	0	-1
2	0	-2
1	0	-1

GRADIENTE COLUMNA

-1	-2	-1
0	0	0
1	2	1

Hay que hacer notar que estrictamente la aplicación del operador de Sobel requiere el uso de un factor de escala $1/4$.

En lo que respecta a los filtros de Frei-Chen las matrices que realizan la operación gradiente son las siguientes:

GRADIENTE FILA

1	0	-1
$\sqrt{2}$	0	$-\sqrt{2}$
1	0	-1

GRADIENTE COLUMNA

-1	$-\sqrt{2}$	-1
0	1	0
1	$\sqrt{2}$	1

Si nos fijamos en la primera y segunda de las máscaras (gradiente de las filas) veremos como resultan de la implementación directa de la fórmula (1.7) y (1.8) que vimos con anterioridad.

La magnitud y orientación del vector gradiente se obtiene de acuerdo a las ecuaciones (1.5) y (1.6), respectivamente; si bien, dado el elevado coste computacional que conlleva el cálculo de la magnitud, ésta suele aproximarse por la expresión 1.9:

$$|G(i, j)| = \sqrt{[G_F]^2 + [G_C]^2} \approx |G_F(i, j)| + |G_C(i, j)| \quad (1.9)$$

Sin embargo, más utilizados que los operadores estudiados en (1.5) y (1.6) son los de Roberts, Prewitt y Sobel, que a continuación se comentan.

La idea del operador de Roberts (Roberts, 1965) es obtener una buena respuesta ante bordes diagonales a partir de la diferencia entre pares diagonales de píxeles. Nótese que , en este caso, las máscaras

no representan las derivadas a lo largo de filas y columnas, sino en dos direcciones diagonales (45°) perpendiculares entre sí. Aunque esta circunstancia no afecta a la expresión para determinar el módulo del gradiente, sí afecta a la orientación, que en este caso viene dada por la expresión:

$$\alpha(i, j) = \pi/4 + \text{atan} [G_2(i, j) / G_1(i, j)] \quad (1.10)$$

donde

$$G_1(i, j) = F(i, j) - F(i+1, j+1) \quad (1.11)$$

$$G_2(i, j) = F(i, j) - F(i+1, j)$$

El operador de Roberts ofrece unas buenas prestaciones en cuanto a localización. El gran inconveniente es su extremada sensibilidad al ruido y, por tanto, con unas pobres cualidades de detección. Los operadores de Prewitt, Sobel y Frei-Chen intentan aliviar esta deficiencia involucrando en la convolución un mayor número de píxeles del entorno de vecindad. Estos tres operadores

pueden formularse de forma conjunta mediante las máscaras de convolución mostradas en la **figura 1.17**. Hay que hacer notar que los gradientes de fila y columna están normalizados para proporcionar una ganancia unidad sobre la zona de pesos positivos y negativos.

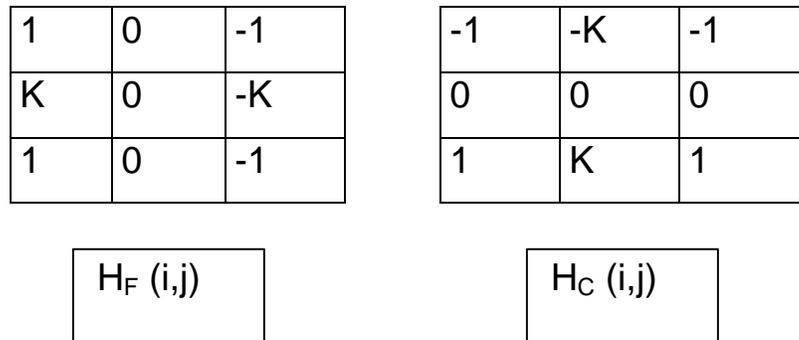


Figura 1.17.- Estructura de los operadores de Prewitt, Sobel y Frei-Chen

donde tanto la matriz de la izquierda como la de la derecha se premultiplica por $1/(2+K)$.

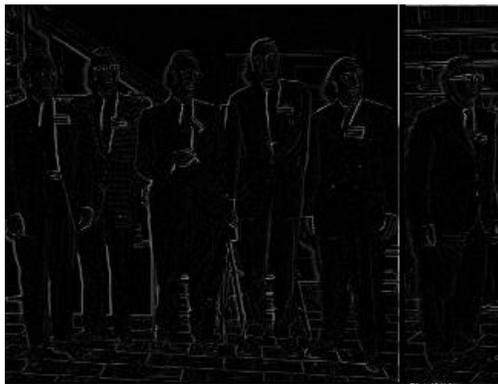
El operador de **Prewitt** ($K = 1$) es, en cierta forma, similar al de diferencia de píxeles separado introducido anteriormente, donde se involucran a los vecinos de filas/ columnas adyacentes para proporcionar mayor inmunidad al ruido.

El operador de **Sobel** se distingue del operador de Prewitt en que se duplican los valores de los píxeles al norte, sur, este y oeste, es decir, $K=2$. De esta forma, cada píxel del entorno de vecindad es ponderado de acuerdo con la distancia de éste al central (píxel evaluado). Como consecuencia de ello, el filtro de Sobel es más

sensible que el de Prewitt a los bordes diagonales, mientras que éste lo es para los horizontales y verticales. En la práctica, sin embargo, no se aprecia gran diferencia entre ellos.

Por otra parte **Frei y Chen** propusieron un valor de $K=\sqrt{2}$ con el fin de que el gradiente sea el mismo para bordes horizontales, verticales y diagonales.

Estos operadores 3x3 tienen unas prestaciones bastante similares. Particularmente, los dos primeros (Prewitt y Sobel) son muy populares, debido a que los valores de los coeficientes hacen que las operaciones aritméticas se puedan programar de una manera eficiente o implementar fácilmente en hardware.



Roberts



Sobel

Figura 1.18.- *Imágenes gradiente obtenidas tras aplicar los operadores de Robert y Sobel a una imagen.*

En la **figura 1.18** se muestra el resultado de aplicar los operadores de Robert y Sobel a una imagen. A continuación procederemos a su justificación matemática.

4.3.2 JUSTIFICACIÓN DE ALGUNOS OPERADORES

4.3.2.1 JUSTIFICACIÓN DEL OPERADOR DE PREWITT

Sea $Z = f(x,y)$ una función de dos variables reales que representa una superficie en R^3 . El plano tangente a Z en un punto (u, v) viene dado por:

$$\begin{aligned} Z(x, y) &= f(u, v) + f_x(u, v)(x - u) + f_y(u, v)(y - v) \\ &= f(u, v) + \nabla f(u, v) \begin{pmatrix} x - u \\ y - v \end{pmatrix} \end{aligned}$$

En el caso discreto la noción de plano tangente no es aplicable ya que la imagen $F(i, j)$ se define por puntos discretos de una malla. En su lugar, se busca un plano $Z(i, j)$ que se ajuste lo mejor posible a los valores discretos de intensidad de los píxeles. Así, para un píxel dado (k,l) , el problema es encontrar un plano que cumpla dos condiciones:

- $Z(k, l) = F(k, l)$
- El plano es el que, según un determinado criterio, mejor se ajusta a la imagen F en el entorno de vecindad del píxel (k, l) .

Sea $Z(i, j) = F(k, l) + A(i - k) + B(j - l)$ el plano buscado, donde A y B son dos parámetros a determinar (componentes del vector que define el plano). Considerando el criterio de mínimos cuadrados según el cual la suma de los cuadrados de la diferencia entre F y Z se minimiza sobre un entorno de vecindad de (k, l) , el problema se plantea como:

$$\text{Min}_{A,B} Q = \min_{A,B} \sum [F(i, j) - Z(i, j)]^2$$

$$(i, j) \in \mathfrak{N}(k, l)$$

donde $\mathfrak{N}(k, l)$ es un entorno de vecindad del píxel (k, l) .

Desarrollando Q para el caso en el que $\mathfrak{N}(k, l)$ son los 8-vecinos de (k, l) se obtiene:

$$\begin{aligned} Q = & [F(k+1, l+1) - F(k, l) - A - B]^2 + [F(k, l+1) - F(k, l) - B]^2 + \\ & + [F(k-1, l+1) - F(k, l) + A - B]^2 + [F(k-1, l) - F(k, l) + A]^2 + \\ & + [F(k-1, l-1) - F(k, l) + A + B]^2 + [F(k, l-1) - F(k, l) + B]^2 + \\ & + [F(k+1, l-1) - F(k, l) - A + B]^2 + [F(k+1, l) - F(k, l) - A]^2 \end{aligned}$$

Para minimizar $Q(A,B)$:

$$\frac{\partial Q}{\partial A}=0 \Rightarrow A = (1/6)\{ [F(k+1,l+1)+F(k+1,l)+F(k+1,l-1)] - [F(k-1,l+1)+ F(k-1,l)+F(k-1,l-1)] \}$$

$$\frac{\partial Q}{\partial B}=0 \Rightarrow B = (1/6)\{ [F(k+1,l+1)+F(k,l+1)+F(k-1,l+1)] - [F(k+1,l-1)+ F(k,l-1)+F(k-1,l-1)] \}$$

que como puede observarse, salvando el factor de escala, son las expresiones del gradiente de fila y columna de Prewitt en el píxel (k,l) .

4.3.2.2. JUSTIFICACIÓN DEL OPERADOR DE SOBEL

La idea con la que surgió este operador es emplear una estimación del gradiente eficiente computacionalmente y más isotrópica que el popular operador de Roberts (Robert, 1965; Danielsson, 1990).

Considérese un píxel con intensidad “ e “ cuyo entorno de vecindad, dado por sus 8- vecinos, posee los siguientes valores de intensidad:

A	B	C
D	E	F
G	H	I

Se define la magnitud del vector de derivada direccional g como:

$$g = \frac{\langle \text{Diferencia de intensidad} \rangle}{\langle \text{Distancia al vecino} \rangle}$$

La dirección de g vendrá dada por el vector unitario desde el píxel central al vecino considerado.

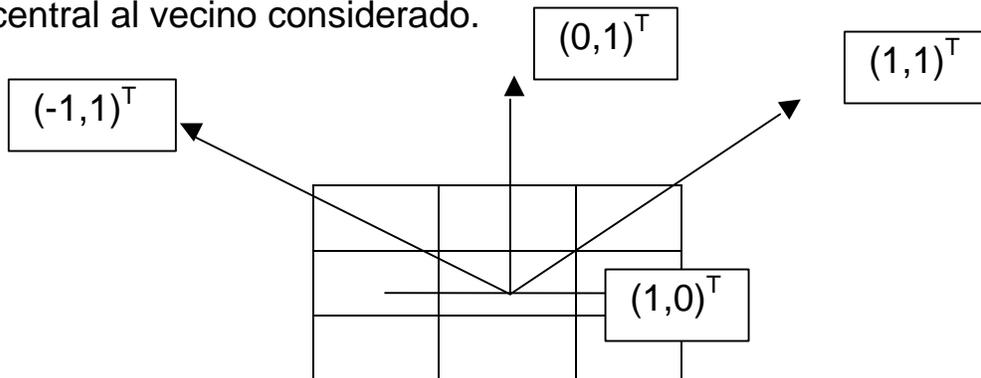


Figura 1.19 .- Direcciones a considerar en las derivadas direccionales g .

El gradiente G en el píxel considerado se estima a partir de la suma de las derivadas direccionales con sus 8-vecinos, es decir:

$$\begin{aligned}
 G &= [(c - e)/R + (e - g)/R](1, 1)^T / R + [(a - e)/R + (e - i)/R](-1, 1)^T / R + \\
 & [(b - e)/R + (e - h)/R](0, 1)^T + [(f - e)/R + (e - d)/R](1, 0)^T = \\
 & = [(c - g)/R](1, 1)^T / R + [(a - i)/R](-1, 1)^T / R + [(b - h)/R](0, 1)^T + \\
 & + [(f - d)/R](1, 0)^T
 \end{aligned}$$

donde $R = \sqrt{2}$.

Operando en la expresión anterior se llega a la siguiente forma simplificada:

$$G = \begin{pmatrix} (c - g - a + i)/2 + f - d \\ (c - g + a - i)/2 + b - h \end{pmatrix} \quad (1.12)$$

Hay que hacer notar, que el término $\sqrt{2}$ desaparece del vector G. Para evitar trabajar con divisiones es conveniente escalar la expresión anterior multiplicando por 2, con lo que se obtiene:

$$G = \begin{pmatrix} (c - g - a + i) + 2(f - d) \\ (c - g + a - i) + 2(b - h) \end{pmatrix} \quad (1.13)$$

Puesto que la expresión (1.12) debería escalarse por el factor $\frac{1}{4}$ para tener ganancia unidad sobre la parte positiva y negativa del operador, la **ecuación 1.13** proporciona en realidad un valor 8 veces superior al gradiente real. Expresando **1.13** en forma de máscaras de convolución se obtienen los gradientes de Sobel expuestos con anterioridad.

Nótese que, este desarrollo del operador de Sobel es equivalente a utilizado para justificar el operador de Prewitt, si bien en este caso, el criterio de mínimos cuadrados se formularía como :

$$\text{Min}_{A,B} Q = \min_{A,B} \sum_{(i,j) \in \mathfrak{X}(k,l)} [F(i,j) - Z(i,j)]^2 / D^2$$

donde D es una función distancia que vale 1 si (i,j) es uno de los cuatro vecinos de (k,l) ó $\sqrt{2}$ si (i,j) es uno de los vecinos diagonales.

4.3.3 IMPLEMENTACIÓN

En la **figura 1.20** se muestra la estructura de un sistema de detección de bordes basado en el cálculo del gradiente. Una vez obtenido el módulo del gradiente en cada píxel (imagen gradiente, $G(i,j)$) suele aplicarse una operación de umbralización, donde se decide si el píxel pertenece o no a un borde de la imagen (imagen binaria de bordes).

La elección del umbral es crítica: un umbral pequeño detectaría como borde fluctuaciones debida al ruido y daría lugar a respuestas múltiples (bordes de espesor mayor que un píxel); por el contrario, un umbral elevado no detectaría muchos de los bordes en la imagen. Lo más adecuado, por tanto, sería emplear un umbral variable en cada píxel, función del nivel de ruido y de la amplitud de la discontinuidad (relación señal-ruido). En la práctica, esto presenta serios inconvenientes que desaconsejan su utilización, como el elevado coste computacional que acarrea y la necesidad de contar con un adecuado modelo probabilístico.

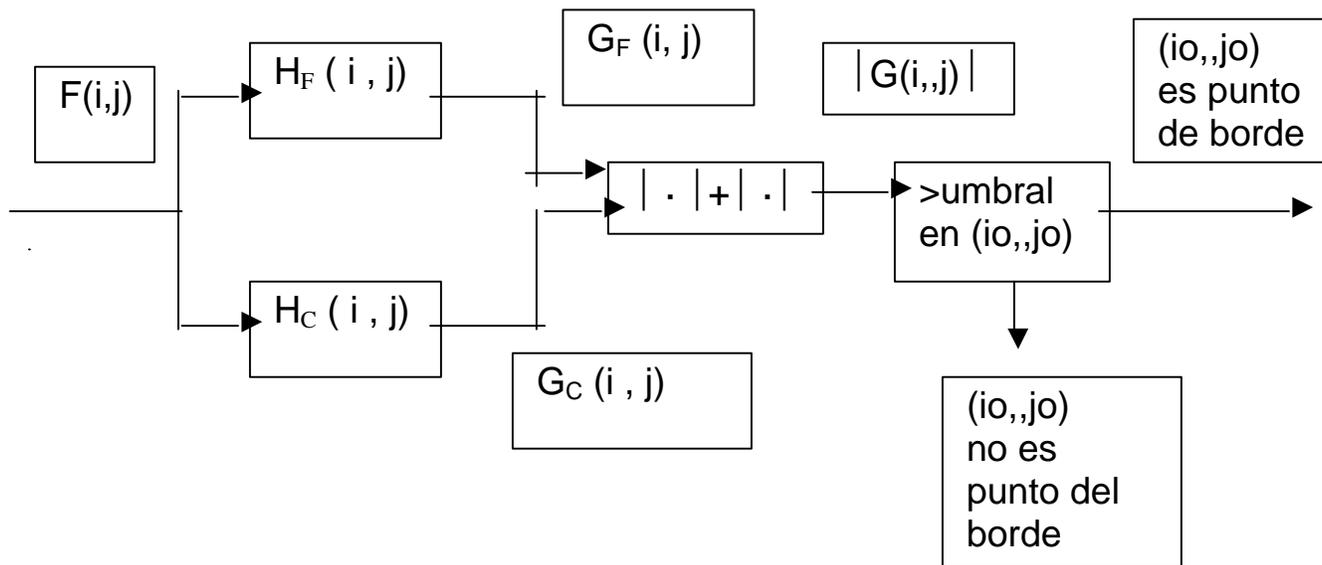


Figura 1.20 .- *Detección de bordes basado en gradiente*

Como ya se indicó, la dirección del gradiente en un píxel de la imagen indica la máxima variación de los niveles de grises, es decir, es perpendicular al borde en dicho punto. Aunque en el sistema mostrado en la **figura 1.20** no se ha tenido en cuenta esta dirección, es importante señalar que ésta suministra una información muy valiosa que puede ser utilizada en posteriores etapas en el proceso de visión (por ejemplo, para la segmentación de objetos mediante seguimiento de contornos y transformada de Hough).

4.3.4 INFLUENCIA DEL TAMAÑO DEL OPERADOR

El tamaño del operador tiene una gran importancia en lo que concierne a sus características de detección y localización. Los operadores presentados hasta ahora (2x2 y 3x3) son muy sensibles a la presencia de ruido en la imagen (en especial los de tamaño 2x2) y,

por consiguiente, con un comportamiento deficiente en cuanto a la detección. Este problema puede mejorarse utilizando máscaras de mayor tamaño que actúan promediando sobre un mayor número de píxeles del entorno. Como se verá en el siguiente apartado, el efecto es similar al que se obtendría aplicando el operador gradiente sobre una imagen previamente suavizada.

Las contrapartidas de utilizar máscaras más grandes son:

- El suavizado de la imagen conlleva la atenuación y enturbiamiento de los bordes y, por consiguiente, una peor localización de éstos.
- El coste computacional es mucho mayor que en operadores pequeños ($O(N \times N)$, siendo N el tamaño de la máscara).

Como regla general, si el operador es pequeño la localización es más precisa, aunque es más sensible al ruido y por ello detectará bordes que no existen. Si el operador es grande entonces la localización es menos precisa, sin embargo, disminuye la sensibilidad al ruido y mejora la detección.

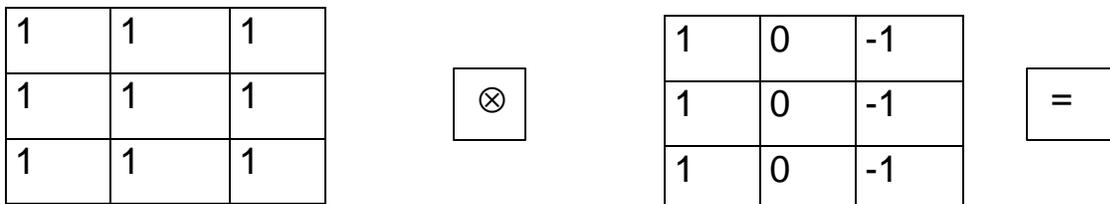
4.3.5 COMBINADO DE SUAVIZADO Y GRADIENTE. OPERADOR $D_{\sigma}G$

Un procedimiento que se puede utilizar para hacer que el operador gradiente sea menos vulnerable al ruido consiste en suavizar previamente la imagen. Ambas operaciones (suavizado y gradiente) se pueden combinar en una sola mediante la convolución:

$$H(i,j) = H_G(i,j) \otimes H_S(i,j)$$

donde $H_S(i,j)$ es la respuesta impulsional de un filtro paso-baja (suavizado), y $H_G(i,j)$ es uno de los operadores gradiente ilustrados con anterioridad.

Para ilustrar todo esto y que sirva a modo de ejemplo cogemos como $H_G(i,j)$ el operador gradiente de fila de Prewitt, y $H_S(i,j)$ el promediado uniforme de 8-vecinos, entonces, el operador resultante será:



1	1	0	-1	-1
2	2	0	-2	-2
3	3	0	-3	-3
2	2	0	-2	-2
1	1	0	-1	-1

La máscara 5x5 resultante suele normalizarse con el factor 1/18 y realiza el gradiente por filas. De esta manera queda patente la afirmación hecha en el apartado anterior acerca de la relación entre el tamaño del operado gradiente y el efecto de suavizado que ello conlleva.

Un operador que combina suavizado y gradiente de una manera más eficaz es el denominado operador DroG o Derivada de la Gaussiana (“ Derivative of Gaussian”), el cual permite, mediante el parámetro de desviación típica de la Gaussiana, controlar el grado de suavizado deseado.

En el caso bidimensional, la Gaussiana isotrópica de desviación típica σ se define como:

$$\begin{aligned} g_{\sigma}(x,y) &= [1/(2\pi\sigma^2)]\exp[-(x^2 + y^2)/2\sigma^2] = [1/2\pi\sigma^2]\exp[-r^2/(2\sigma^2)] = \\ &= g_{\sigma}(r) \text{ con } r=(x^2 + y^2)^{1/2} \end{aligned} \quad (1.14)$$

El cálculo del gradiente sobre una función continua $f(x, y)$ suavizada con $g_{\sigma}(x, y)$ viene dado por la composición de operaciones:

$$\nabla [f(x,y) \otimes g_{\sigma}(x, y)] \quad (1.15)$$

donde $\nabla [\bullet]$ denota el operador gradiente.

Puesto que, tanto el gradiente como la convolución son operaciones lineales, la ecuación 1.15 puede ponerse como:

$$\nabla [f (x,y) \otimes g_{\sigma} (x, y)] = f (x, y) \otimes \nabla [g_{\sigma} (x, y)] = f (x, y) \otimes \text{DroG}(x,y)$$

El operador DroG es un vector que se define como:

$$\begin{aligned} \text{DroG}(x, y) &= \left\{ \begin{array}{c} \partial [g_{\sigma}(x)g_{\sigma}(y)]/\partial x \\ \partial [g_{\sigma}(x)g_{\sigma}(y)]/\partial y \end{array} \right\} = \\ &= \begin{pmatrix} g_{\sigma}(y)\partial [g_{\sigma}(x)]/\partial x \\ g_{\sigma}(x)\partial [g_{\sigma}(y)]/\partial y \end{pmatrix} = \begin{pmatrix} -x g_{\sigma}(x)g_{\sigma}(y)/\sigma^2 \\ -y g_{\sigma}(x)g_{\sigma}(y)/\sigma^2 \end{pmatrix} \end{aligned} \quad (1.16)$$

donde se ha tenido el carácter separable de la Gaussiana, es decir:

$$g(x, y) = g_{\sigma}(x) g_{\sigma}(y)$$

Hay que hacer notar, que puesto que el operador DroG sólo se ha de calcular una sola vez, en lugar de realizar dos operaciones distintas sobre la imagen (primero la Gaussiana y después la derivada), sólo se tiene que realizar una convolución.

Para implementar el operador DroG en una imagen digital es necesario discretizar los componentes “x “ e “ y “ de la ecuación 1.16. Esta discretización se realiza de acuerdo con el valor de la desviación típica σ , teniendo en cuenta las mismas consideraciones que en el caso del operador de suavizado gaussiano descrito con anterioridad, obteniéndose dos máscaras, una para el gradiente de filas y otra para el gradiente de columnas. Obsérvese que el operador DroG se aplica, de acuerdo con la ecuación 1.16, convolucionando el vector derivada de la gaussiana con la imagen una vez suavizada con el vector gaussiana.

Por último, hay que hacer hincapié que el resultado que se obtiene con DroG es menos sensible al ruido que el obtenido con el operador de Sobel.

4.4 OPERADORES BASADOS EN LA SEGUNDA DERIVADA (LAPLACIANA).

Siempre que en la imagen se presenta un cambio de intensidades a lo largo de una determinada dirección, existirá un máximo en la primera derivada a lo largo de dicha dirección y, consecuentemente, un paso por cero en la segunda derivada. La dirección de interés será aquella ortogonal a la orientación local de los cruces por cero.

Puesto que, generalmente esta dirección no es conocida, parece claro que la utilización de un operador independiente de la orientación como la Laplaciana presenta evidentes ventajas computacionales.

Asimismo, al ser una derivada segunda, tiene la ventaja de facilitar la localización precisa del borde.

En el dominio continuo la laplaciana del borde de una función bidimensional $f(x, y)$ se define como:

$$\nabla^2 f(x, y) = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) f(x, y) \quad (1.17)$$

La Laplaciana vale cero si $f(x, y)$ es constante o cambia linealmente su amplitud. El cambio de signo de la función resultante nos indica que en ese lugar existe un cruce por cero (cambio de positivo a negativo o viceversa) y, por tanto, indica la presencia de un borde. Hay que hacer constar que, a diferencia del gradiente, la Laplaciana no es un vector.

4.4.1 IMPLEMENTACIÓN DE LA LAPLACIANA

En el dominio discreto la Laplaciana se aproxima mediante la diferencia de pendientes a lo largo de cada uno de los ejes. Supóngase que la derivada parcial $\frac{\partial f(x, y)}{\partial x}$ se aproxima mediante:

$$f_x(x, y) = G_F(i, j) = f(i+1, j) - f(i, j)$$

donde una vez más se ha omitido el factor de escala ya que no afecta al cruce por cero. La segunda derivada a lo largo del eje de filas, $f_{xx}(x, y)$, se aproxima mediante:

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial f_x}{\partial x} = G_F(i+1, j) - G_F(i, j) = f(i+1, j) - 2f(i, j) + f(i-1, j) \quad (1.18)$$

Procediendo de igual manera para la segunda derivada en el eje “y” (columnas) se tiene:

$$\frac{\partial^2 f}{\partial y^2} = \left(\frac{\partial}{\partial y}\right) f_y = G_C(i, j+1) - G_C(i, j) = f(i, j+1) - 2f(i, j) + f(i, j-1) \quad (1.19)$$

Por tanto, la aproximación discreta de la Laplaciana de la imagen $f(i, j)$ se obtiene mediante la convolución:

$$L[F(i, j)] = F(i, j) \otimes H(i, j) \quad (1.20)$$

donde:

$f(i, j) =$	0	0	0	+	0	-1	0	=
	-1	2	-1		0	2	0	
	0	0	0		0	-1	0	

0	-1	0
-1	4	-1
0	-1	0

Esta Laplaciana se conoce como Laplaciana de 4 vecinos, la cual suele normalizarse para que tenga una ganancia unidad sobre los píxeles de pesos positivos y negativos, es decir:

$$H(i,j) = \frac{1}{4} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

También se puede definir una Laplaciana normalizada de 8 vecinos:

$$H(i,j) = \frac{1}{8} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Aunque ésta no es separable como sumas de segunda derivadas como en las **ecuaciones 1.18 y 1.19**.

La principal característica de la Laplaciana es la capacidad de localizar los bordes de forma precisa mediante la determinación del cruce por cero. En la **figura 1.21** se muestra el diagrama de detección típico de un sistema de detección de bordes basado en la Laplaciana.

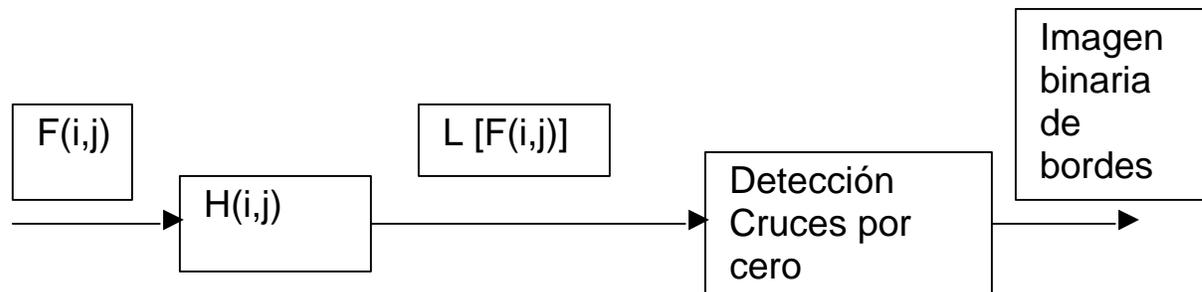


Figura 1.21 .- *Sistema para la detección de bordes basado en la Laplaciana.*

Hay que darse cuenta que, al ser las imágenes discretas, es muy poco probable que el paso por cero coincida en un píxel de la imagen. Por ello, el borde se suele marcar en aquel píxel cuyo nivel de gris es más próximo a cero, teniendo como vecino, al menos, un píxel con nivel de gris de signo contrario (Huertas y Medioni, 1986).

Las características principales de la Laplaciana son :

- Al ser un operador de segunda derivada, exhibe una pobre detección de bordes ya que es muy sensible al

ruido. Para evitarlo, en lugar de aplicar la Laplaciana directamente, se realiza primero un suavizado.

- Si los bordes de la imagen original $F(i, j)$ se verifican ciertas condiciones de suavidad, los cruces por cero de la imagen Laplaciana resultante son curvas cerradas (Torre y Poggio, 1986).
- Proporciona un borde de espesor unidad (un píxel).

4.4.2 COMBINADO DE SUAVIZADO Y LAPLACIANA. OPERADOR LOG

Como se acaba de comentar, el operador Laplaciana es muy sensible al ruido, por lo cual es prácticamente imprescindible su utilización de forma combinada con un operador de suavizado. En particular Marr y Hildreth (1980) propusieron el operador LoG donde se aplica la Laplaciana a la imagen una vez suavizada con un filtro gaussiano. Recurriendo, una vez más, al dominio continuo se tiene:

$$\nabla^2 [f(x,y) \otimes g_\sigma(x,y)] = f(x,y) \otimes \nabla^2 [g_\sigma(x,y)] = f(x,y) \otimes \text{LoG}_\sigma(x,y)$$

donde $g_\sigma(x,y) = g_\sigma(x) g_\sigma(y)$ es la respuesta impulsional de una función gaussiana bidimensional.

Aplicando la expresión de la gaussiana se tiene:

$$\begin{aligned} \text{LoG}_\sigma(x,y) &= [1/(\pi\sigma^4)] * [(x^2 + y^2) / (2\sigma^2) - 1] e^{-[(x^2 + y^2) / (2\sigma^2)]} = \\ &= 1/(\pi\sigma^4) [r^2 / 2\sigma^2 - 1] e^{-r^2 / 2\sigma^2} = \text{LoG}_\sigma(r) \end{aligned}$$

Por lo tanto, basándonos en la expresión anterior el operador LoG presenta simetría radial adoptando una forma parecida a la de un sombrero mejicano. Por este motivo, este operador se conoce con el nombre de “ filtro de sombrero mejicano”.

Es importante observar cómo la combinación de un filtro paso-alta (Laplaciana) y un filtro paso-baja (Gaussiana) da lugar a un filtro paso-banda, cuyo ancho de banda es controlado por el parámetro σ de desviación típica.

Aunque el operador LoG no es separable, en virtud de la separabilidad de la Gaussiana, éste puede ponerse como suma de operadores separables (Huertas y Medioni, 1986).

$$\text{LoG}_\sigma(x,y) = 1/[\pi\sigma^2] g_\sigma(x) [y^2/\sigma^2 - 1] g_\sigma(y) + 1/(\pi\sigma^2) g_\sigma(y) [x^2/\sigma^2 - 1] g_\sigma(x)$$

Teniendo en cuenta el gran tamaño requerido para este operador, este hecho tiene una gran trascendencia desde el punto de vista de la eficiencia computacional, puesto que permite efectuar la convolución por filas y columnas de la imagen de manera separada, reduciendo el número de operaciones de n^2 a $4n$, siendo n el tamaño del operador.

En vez de utilizar este operador aproximaremos LoG mediante un operador diferencia de gaussianas denominado DoG.

Una alternativa muy usada consiste en aproximar LoG por un operador diferencia de gaussianas denominado DoG. El operador DoG para las desviaciones típicas σ_1 y σ_2 se define como:

$$\text{DoG}_{\sigma_1 \sigma_2}(x,y) = g_{\sigma_1}(x,y) - g_{\sigma_2}(x,y) = g_{\sigma_1}(x)g_{\sigma_1}(y) - g_{\sigma_2}(x)g_{\sigma_2}(y)$$

Más concretamente, Marr y Hildreth en 1980 demostraron que la proporción $\sigma_2 / \sigma_1 = 1.6$ es la que mejor aproxima la forma del operador LoG.

Al igual que con la Laplaciana, los cruces por cero obtenidos tras aplicar el operador Laplaciana de la Gaussiana suelen producir contornos cerrados, mas aún en este caso por el efecto de suavizado inherente al propio operador. Esta propiedad puede resultar muy interesante para la posterior segmentación de la imagen.

Aunque existen algoritmos, más o menos sofisticados, capaces de determinar la localización de los cruces por cero con precisión de subpíxel (Huertas y Medioni, 1986), éstos pueden ser detectados de una manera simple de la siguiente forma: un píxel es marcado como cruce por cero si su valor en la imagen resultante de aplicar el operador LoG es menor que $-t$ y uno de sus vecinos es mayor que t , o también si su valor es mayor que t y uno de sus vecinos es menor que $-t$, siendo t un umbral positivo (Pratt, 1991).

El principal inconveniente del operador LoG, además del elevado coste computacional que conlleva, es su pobre comportamiento en

regiones donde intersectan bordes con distintas orientaciones, deformándose las esquinas y desconectándose bordes. Estos defectos son analizados en detalle en (Marr,1980; Marr y Hildreth,1980; Torre y Poggio,1986).

La implementación discreta del operador LoG se realiza muestreando la función continua dada por las expresiones LoG y DoG. Para reducir el efecto de truncado de estas funciones, el tamaño de la máscara debe ser $W \geq 3c$, donde $c = 2\sqrt{2}\sigma$ es el ancho del lóbulo central. En la figura 1.23 se muestra un ejemplo de LoG discreto para $W=11$ y $\sigma=1.4$.

Como se ha comentado, el parámetro σ , y por consiguiente W , controla el grado de suavizado de la imagen y, por tanto, el número de cruces por cero.

0	0	0	-1	-1	-2	-1	-1	0	0	0
0	0	-2	-4	-8	-9	-8	-4	-2	0	0
0	-2	-7	-15	-22	-23	-22	-15	-7	-2	0
-1	-4	-15	-24	-14	-1	-14	-24	-15	-4	-1
-1	-8	-22	-14	52	103	52	-14	-22	-8	-1
-2	-9	-23	-1	103	178	103	-1	-23	-9	-2
-1	-8	-22	-14	52	103	52	-14	-22	-8	-1
-1	-4	-15	-24	-14	-1	-14	-24	-15	-4	-1
0	-2	-7	-15	-22	-23	-22	-15	-7	-2	0
0	0	-2	-4	-8	-9	-8	-4	-2	0	0
0	0	0	-1	-1	-2	-1	-1	0	0	0

TERCERA PARTE. RESULTADOS OBTENIDOS.

1. INTRODUCCIÓN.

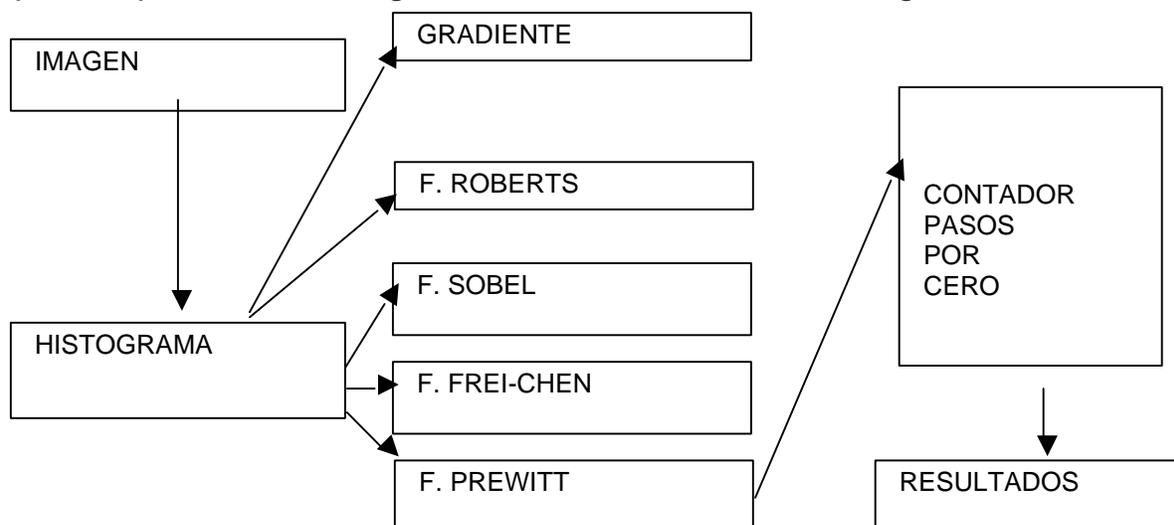
Para analizar cualquier imagen que se nos presente a la vista tenemos que realizar un procesamiento sobre la misma que nos permita eliminar gran parte de la información que nos presenta y que no es relevante para el estudio que vamos a realizar, porque de lo contrario nos encontraríamos no sólo una imagen con mucho ruido sino también con una gran complejidad computacional que ralentizará cualquier cálculo efectuado sobre la misma. Por todo ello hemos planteado una metodología a seguir para la consecución de nuestros objetivos.

En primer lugar realizamos un histograma de la imagen que nos proporciona una representación gráfica de la frecuencia con la que los niveles de grises aparecen en dicha imagen. En el eje de abscisas se indican los distintos niveles (discretos) de grises y en ordenadas se representa la frecuencia o el número de píxeles que poseen ese nivel de gris. Además el histograma nos proporcionará información estadística sobre como están distribuidos los distintos niveles de grises de la imagen. Esta información nos es útil por las siguientes razones:

- Para saber si la digitalización de la imagen se ha realizado correctamente y el nivel de umbral.
- Nos permite establecer una analogía entre el histograma y la función de densidad de probabilidad cuanto mayor sea el número de niveles de intensidad de la imagen.

Sin embargo esto no es suficiente ya que después de realizar este estudio sobre multitud de imágenes vimos la necesidad de depurar las imágenes para obtener más información de ella. Por este motivo, después del estudio antes realizado decidimos aplicar unos filtros de transformación a las imágenes. Como el objetivo era detectar contornos lo que teníamos que detectar eran variaciones bruscas de la imagen. Por ello acudí a calcular el gradiente que nos expresará el resultado de aplicar a la imagen una primera derivada. Una vez realizado dicho procesamiento lo conjugaremos con las técnicas de umbralización antes mencionadas.

Seguidamente nos adentraremos en el estudio del operador de Roberts ya que ofrece buenas prestaciones en cuanto a localización y búsqueda de contornos diagonales de 45° perpendiculares entre sí. Como consecuencia del resultado obtenido nos obliga a adentrarnos en la búsqueda de otros filtros que nos proporcionen mayores prestaciones. Los filtros que darán razón de ser a la siguiente parte del estudio serán los filtros de Sobel, Prewitt y Frei-Chen que involucran mayor número de píxeles de la vecindad en el procesamiento de una imagen. A continuación voy a esbozar un gráfico que remarcará los pasos que vamos a seguir en el tratamiento de la imagen.



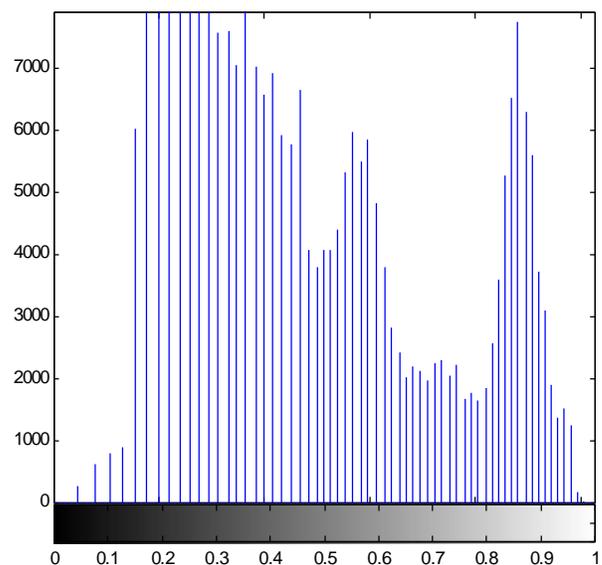
A continuación pasamos a describir los resultados acontecidos en cada etapa.

2. PASOS EJECUTADOS EN EL PROYECTO.

2.1 HISTOGRAMA

Me planteé esta etapa como una necesidad a cubrir antes de emprender siguientes etapas, ya que no podíamos comenzar o no deberíamos comenzar sin haber hecho un estudio preliminar de la imagen, de sus características que nos permita en etapas posteriores abarcar ciertos enfoques en función de ellas. Para abarcar este estudio de la infinidad de imágenes que tenemos hemos optado por la imagen “Gatlin” en la que se muestra una fotografía de 6 personas de cuerpo entero.

En primer lugar he realizado un histograma de la imagen original sin realizar ningún proceso de transformación sobre ellas. A la izquierda presentamos la imagen original y a la derecha el histograma



Del gráfico expuesto observo como la imagen concentra los valores en el intervalo [0.2- 0.6] presentando un valor elevado del mismo en las proximidades del 0.9. Esto es así porque en el eje de ordenadas se representa el número de píxeles que tienen el tono de gris manifestado en abscisas. Una vez que sabemos algo más de la imagen me propongo subir el umbral de la misma, es decir, voy a tomar un valor (umbral) que determinará que todos los puntos por debajo de ese valor tomarán el color negro (píxel=0) y los puntos por encima del mismo tomarán el color blanco (píxel=1). A continuación voy a presentar las distintas imágenes que se obtienen al umbralizar la imagen original.

UMBRAL=0.1



UMBRAL=0.2



UMBRAL = 0.3



UMBRAL = 0.4



UMBRAL = 0.5



UMBRAL = 0.6



UMBRAL = 0.7



UMBRAL = 0.8



UMBRAL = 0.9

Como mencioné anteriormente los valores del histograma estaban concentrados en el intervalo [0.2 – 0.6], pero si lo que queremos centrarnos es en detectar las caras, podemos observar que en la imagen original las caras de los seis miembros presentan unos tonos muy claros y habrá que buscarlas entre esos valores. Por lo que a este análisis respecta el valor de umbral adecuado para detectar la cara esta en 0.6 y 0.7 ya que nos permite tener una imagen de la cara más simplista presentándonos sus contornos más sobresalientes.

A pesar de todo ello esta solución no nos resuelve el problema por las siguientes razones:

- Se esta trabajando sobre una imagen con excesiva información.
- Perdemos definición de los contornos de la imagen ya que hago cero los píxeles de la imagen que están por debajo del umbral.

Sin embargo no debemos ver los resultados obtenidos como un fracaso sino como una oportunidad ya que nos permitirá complementar posteriores estudios.

2.2 OPERADORES GRADIENTE

Al abordar el problema de la detección de contornos en la cara no me podía centrar solamente en la detección de contornos verticales y horizontales sino que también debía encontrar contornos diagonales. Como fruto de esta motivación se planteó la oportunidad de usar el operador de Roberts que permite obtener una buena respuesta ante

bordes diagonales a partir de la diferencia entre pares diagonales de píxeles. En este caso las máscaras que utilizo no representan las derivadas a lo largo de filas y columnas, sino en dos direcciones diagonales (45°) perpendiculares entre sí.

Tras la aplicación del operador de Roberts se obtienen los siguientes resultados:

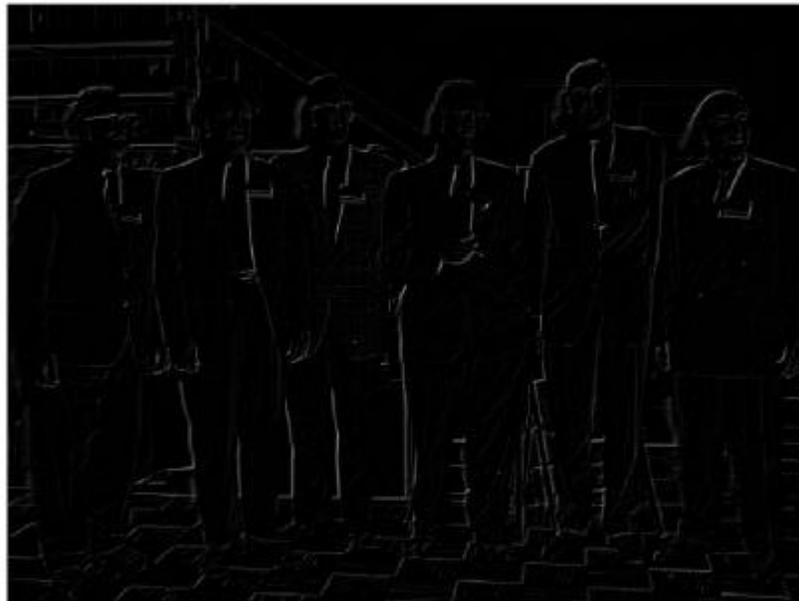
- Haciendo la derivada en la diagonal secundaria inferior se obtiene el resultado mostrado en la figura 1.



- Efectuando la derivada en una dirección perpendicular a la anterior pero formando un ángulo de 135° tenemos:



- Después de los resultados obtenidos y extrayendo lo mejor de los dos filtros aplicados se obtiene :



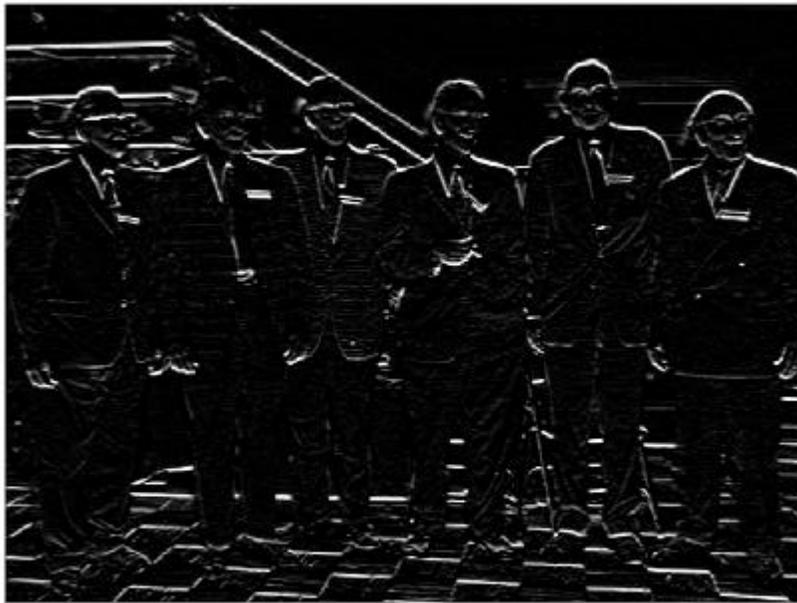
Como se puede apreciar sobre todo si lo comparamos con los resultados que obtuvimos con el histograma, se aprecia menor cantidad de información innecesaria para efectuar nuestro análisis sobre la imagen, lo cual es positivo sin embargo, este método no nos reporta un mínimo requerido para nuestro estudio por las siguientes razones:

- Pobres cualidades de detección.
- Extremada sensibilidad al ruido.

Esto es debido a que el operador de Roberts involucra a un número de píxeles muy pequeño del entorno de vecindad. Para solucionar este problema decidí acudir a los operadores de Sobel, Prewitt y Frei-Chen que intentan aliviar esta deficiencia involucrando en la convolución un mayor número de píxeles del entorno de vecindad.

Por todo lo expresado con anterioridad utilizaré el filtro de Sobel. El operador de Sobel se distingue del operador de Prewitt que después comentaré, en que se duplican los valores de los píxeles de la matriz situados al norte, sur, este y oeste. Con esto se consigue que cada píxel del entorno de vecindad sea ponderado de acuerdo con la distancia de éste al central (píxel evaluado). Aplicando el operador de Sobel a distintas imágenes pero sobretodo centrándonos en la imagen Gatlin se presentan los siguientes resultados que posteriormente pasaré a comentar:

- En primer lugar aplico a la imagen un filtro que detecte los contornos horizontales. El resultado que se obtiene tras ello es el siguiente:



Como se puede apreciar, en la imagen aparecen más resaltados los contornos horizontales que los verticales. Para reafirmar esto nada más hay que ver que los dibujos de las baldosas sólo están resaltados los contornos horizontales.

- En segundo lugar, aplico a la imagen un filtro que detecte los contornos verticales. El resultado de aplicar este filtrado es el siguiente:



Como se puede apreciar, en la figura aparecen más resaltado los contornos verticales que los horizontales. Se aprecia este detalle no sólo en las personas que componen la fotografía sino también en las baldosas.

- A continuación, después del resultado obtenido en los puntos 1 y 2 pasamos a tomar lo mejor del filtrado anterior, es decir, vamos a ir píxel por píxel del resultado de la convolución de la imagen tanto con el filtro que detecta los contornos verticales como el que detecta los contornos horizontales y el valor mayor de cada uno de ellos (el más luminoso) lo iremos depositando en una matriz que es la que a continuación mostraré:

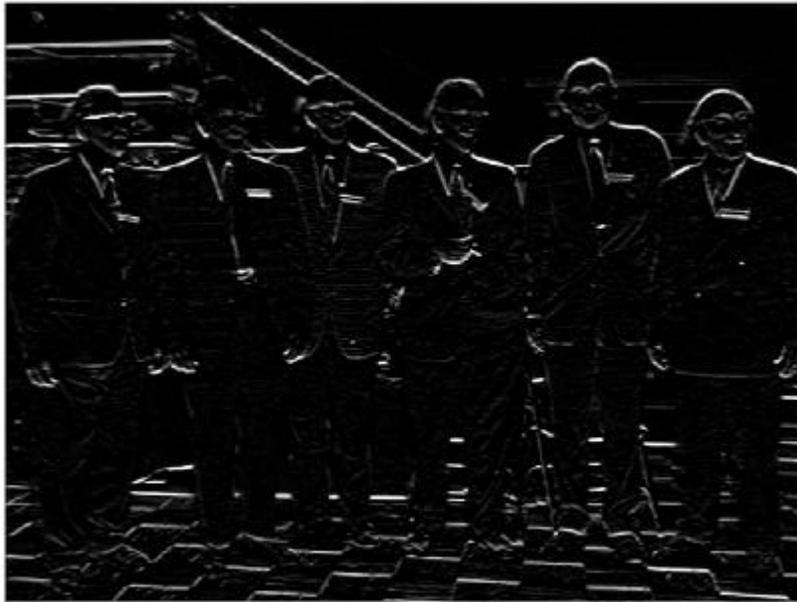


Como se puede apreciar los contornos de la imagen tanto verticales como horizontales aparecen resaltados. Sin embargo, en la zona de la cabeza el realce no es muy acusado y esperamos por

encontrar un filtro que colme nuestras aspiraciones decido acudir al filtro de Prewitt.

El operador de Prewitt es, en cierta forma, similar al de diferencia de píxeles separados, donde se involucran a los vecinos filas/columnas adyacentes para proporcionar mayor inmunidad al ruido. Si aplicamos el operador Prewitt a la imagen Gatlin se obtienen los siguientes resultados:

- Realce de los contornos horizontales de la imagen.



Como se puede apreciar en la imagen predomina el realce de los contornos horizontales frente a los verticales y esto se debe al tipo de matriz que he utilizado para hacer la transformación.

- Realce de los contornos verticales de la imagen.



Del resultado expuesto se puede apreciar un mayor realce de los contornos verticales como antes mencionaba.

- Realce de contornos tanto verticales como horizontales.



Como se puede apreciar tanto con el filtro de Sobel como con el de Prewitt se obtiene un resultado donde se resaltan los contornos mejor que con el filtro de Roberts y con el filtro del gradiente.

Antes de acometer un tratamiento sobre los resultados anteriormente obtenidos describo el último método aplicado a una

imagen. En este caso la imagen original sobre la que se ha realizado el procesamiento es bien distinta de las anteriores. Se trata de la imagen del payaso también conocida como 'clown'. Si bien en los demás casos construí la función que me permitiera realizar cada uno de los filtros anteriormente descritos en este caso me he limitado a usar la función " edge" que viene en la librería cambiándoles los parámetros para aplicarles distintos tipos de filtrado. Los resultados obtenidos son los siguientes:

- EDGE : PARÁMETRO ROBERTS



- EDGE: PARÁMETRO SOBEL



- EDGE: PARÁMETRO PREWITT



- EDGE: PARÁMETRO LOG



- EDGE: PARÁMETRO ZEROCROSS



A raíz de estos resultados observamos como la función edge con el parámetro Roberts obtiene un resultado mucho más pobre que los otros. También hay que hacer mención que el resultado que se obtiene con el parámetro Sobel es mejor que el que se obtiene con el parámetro Prewitt en el sentido que se consigue extraer mayor número de contornos. No obstante los resultados estrella son los que se consiguen con el parámetro “log” y “zerocross” ya que se consiguen marcar muchos detalles de la imagen. No obstante hay que decir, que no es tan bueno tener tantos detalles de la imagen ya que si sobre este resultado vamos a aplicar otro proceso o transformación este exceso de detalles nos hará tener una mayor carga computacional que

ralentizará los cálculos e impedirá ejecutarlos en tiempo real que puede o no ser vital dependiendo de la aplicación para la que se use.

Por último hay que decir que después de aplicar el filtro de Frei-Chen no se obtiene un resultado que mejore los anteriormente obtenidos. A pesar de todo voy a exponer los resultados gráficos que se obtuvieron:



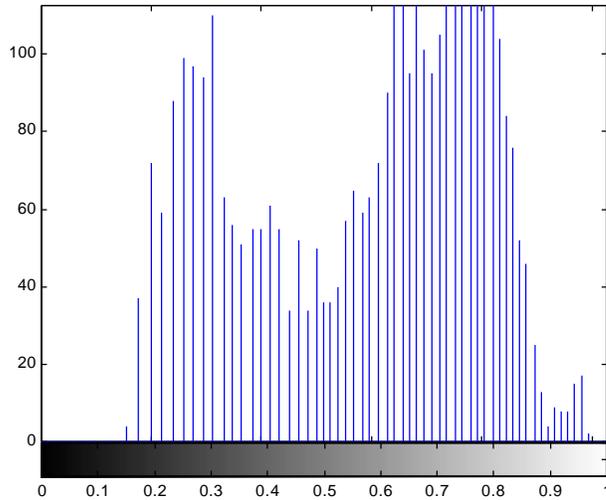


Como se puede apreciar y a modo de balance he de decir que el mejor resultado se obtiene con el filtro de Sobel y el de Prewitt por los siguientes motivos:

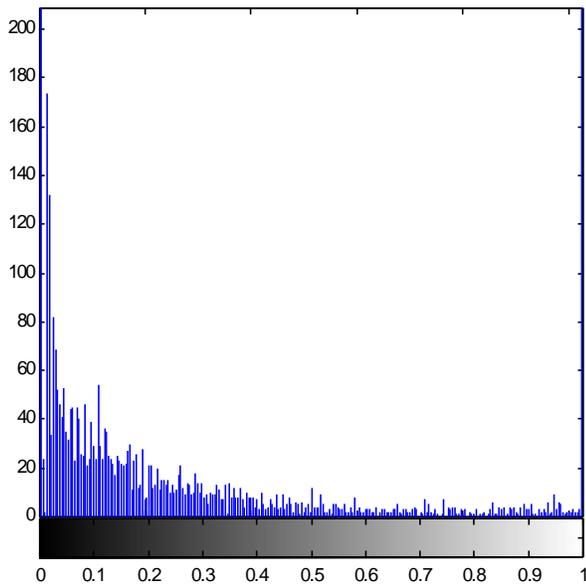
- Realizar cualquier tipo de procesamiento sobre una señal a la que la haya aplicado un filtro de Roberts significa abordar el problema con una carencia de información debida a las pobres cualidades del filtro en cuestión.
- También nos podemos encontrar con el caso contrario como ocurre con la función “edge” que al aplicarle los parámetros “zerocross” o “log” me crea una imagen con muchísimos detalles pero que ralentiza cualquier procesamiento ulterior sobre la misma.

Después de todo el estudio realizado y profundizando en el mismo abordamos la idea de encontrar zonas características de la cara con el objeto de hacer una medición entre dichas zonas para así poder caracterizar una imagen mediante la inclusión de dichas medidas en una base de datos relacional que nos permita establecer comparaciones entre las medidas de una persona y el resto con objeto de identificarla. Para realizar este estudio acudimos al filtro que nos dio un mejor resultado en la etapa anterior, que fue el de Prewitt. Hay que tener en cuenta que las características del histograma sobre la imagen original son bien distintas de las características del histograma sobre la imagen filtrada. Así, en primer lugar lo que voy a hacer es

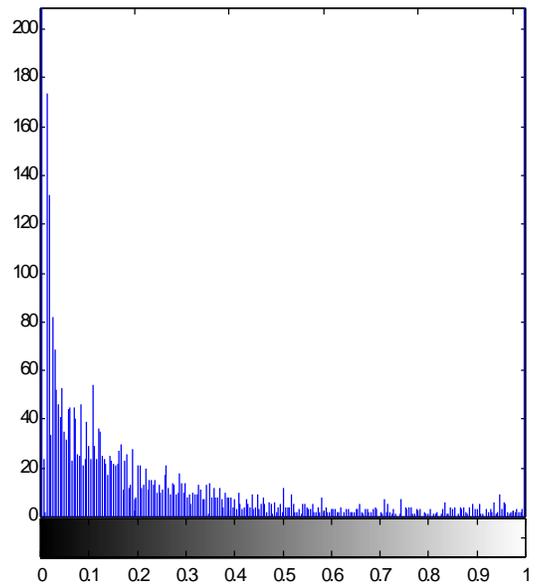
exponer el resultado de esos histograma sobre la cabeza antes y después de filtrar. Los resultados son los siguientes:



HISTOGRAMA DE LA 1ª CABEZA SIN FILTRAR



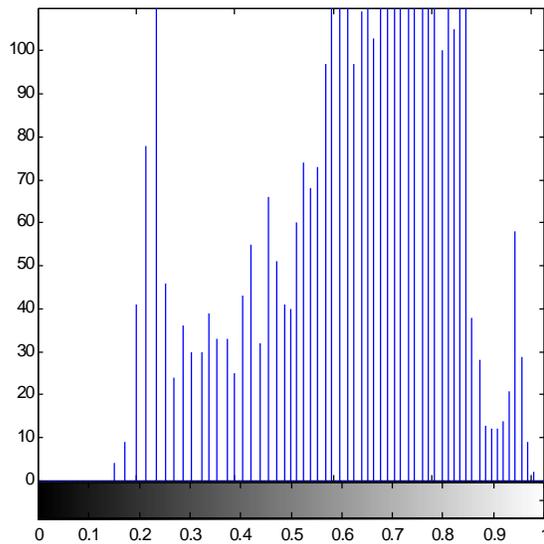
HISTOGRAMA DE LA 1ª CABEZA APLICANDO F. PREWITT



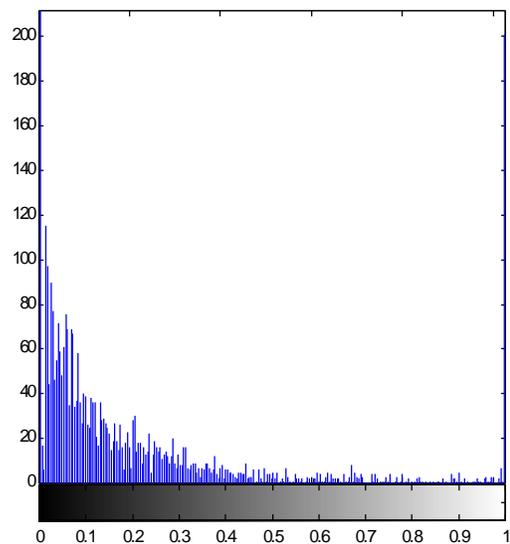
HISTOGRAMA DE LA 1ª CABEZA APLICANDO F. SOBEL

Este trabajo se ha efectuado sobre todas y cada una de las cabezas de la imagen Gatlin con objeto de verificar similitudes en las distintas zonas de la cara.

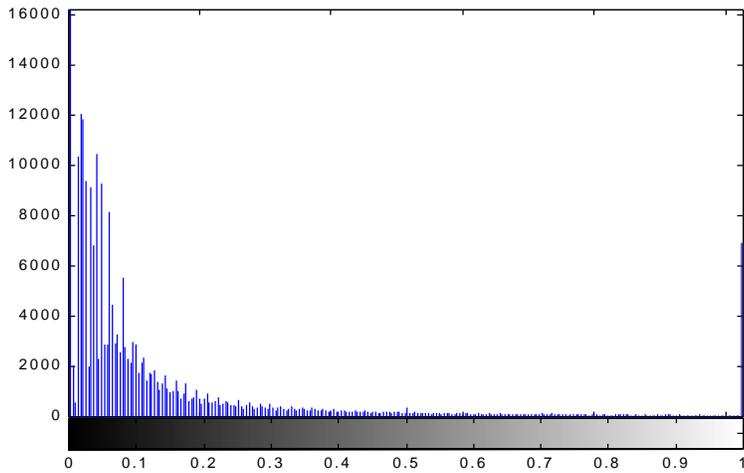
Los resultados que se han obtenido para la segunda imagen de Gatlin empezando por la izquierda son los siguientes:



HISTOGRAMA
DE LA 2ª CABEZA
SIN FILTRAR

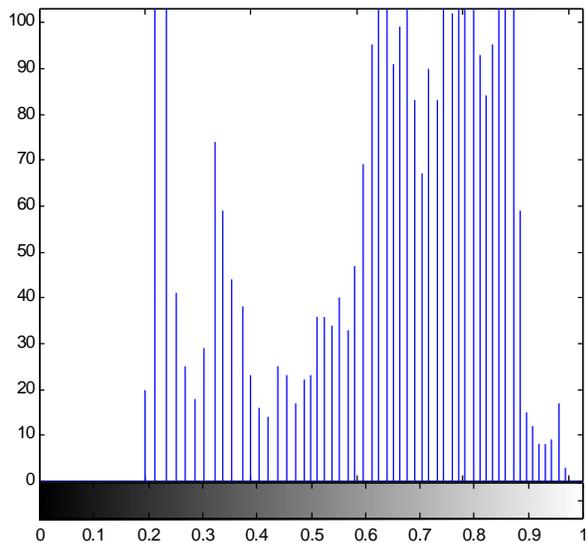


HISTOGRAMA
DE LA 2ª CABEZA
UNA VEZ APLICADO
F. PREWITT

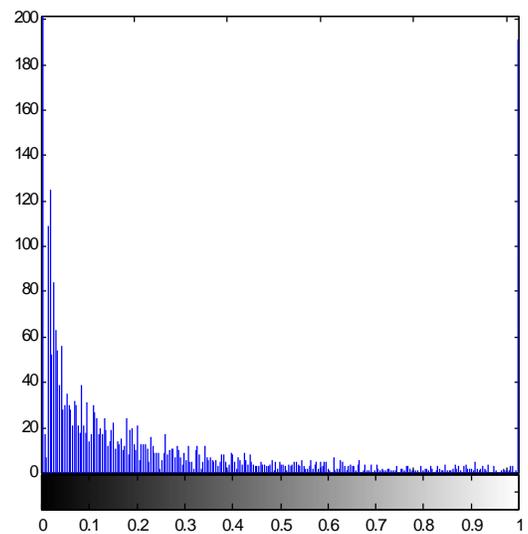


HISTOGRAMA
DE LA 2ª CABEZA
DEPUÉS DE APLICAR
F. SOBEL

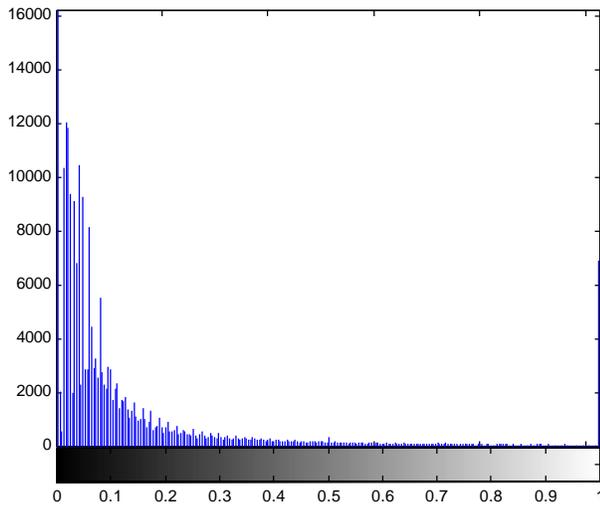
Los resultados que se han obtenido al realizar el histograma antes y después del filtrado son los siguientes:



HISTOGRAMA
DE LA 3ª CABEZA
SIN FILTRAR

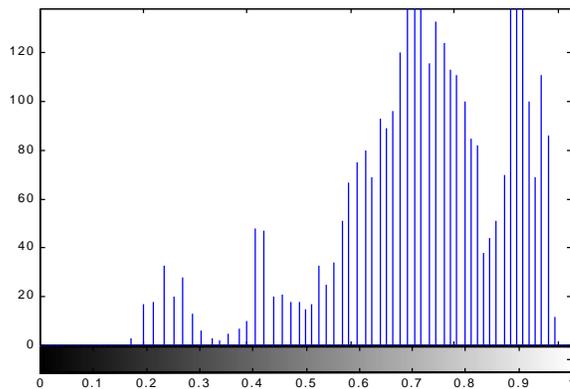


HISTOGRAMA
DE LA 3ª CABEZA
APLICANDO
F. PREWITT

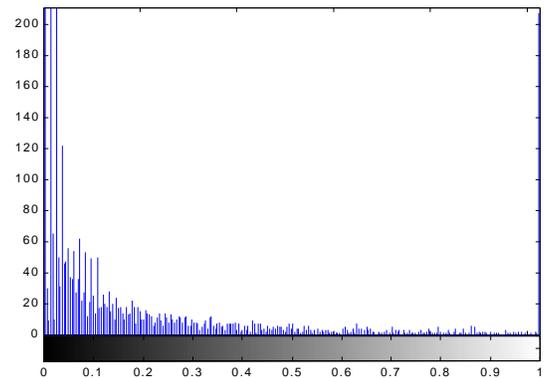


HISTOGRAMA DE LA 3ª CABEZA APLICANDO F. SOBEL

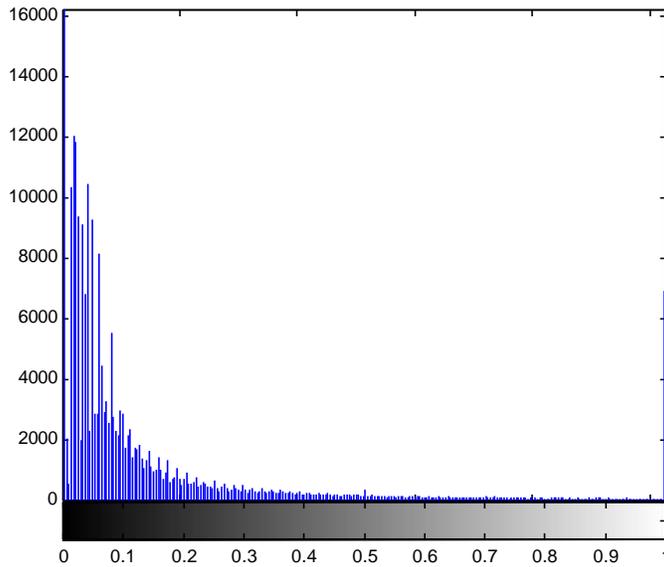
Los resultados que se han obtenido al realizar el histograma antes y después del filtrado sobre la 4ª cabeza de la imagen Gatlin son los siguientes:



HISTOGRAMA DE LA 4ª CABEZA SIN TRANSFORMAR

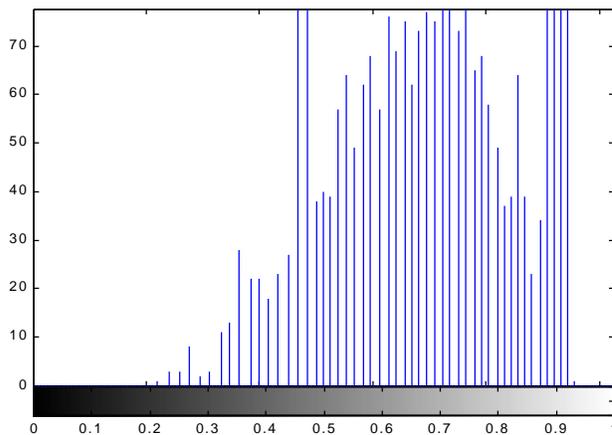


HISTOGRAMA DE LA 4ª CABEZA DESPUÉS DE APLICAR F. PREWITT

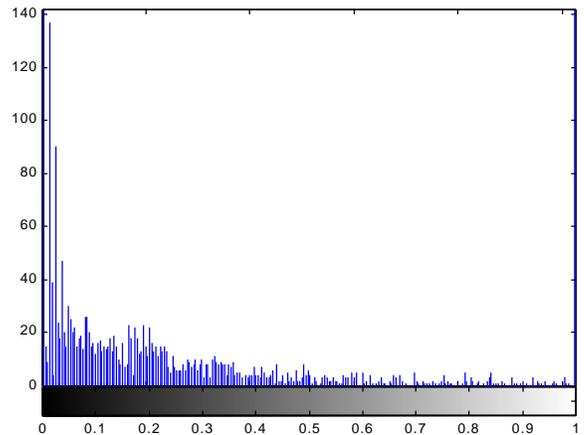


HISTOGRAMA DE LA 4ª CABEZA DESPUÉS DE APLICAR SOBEL

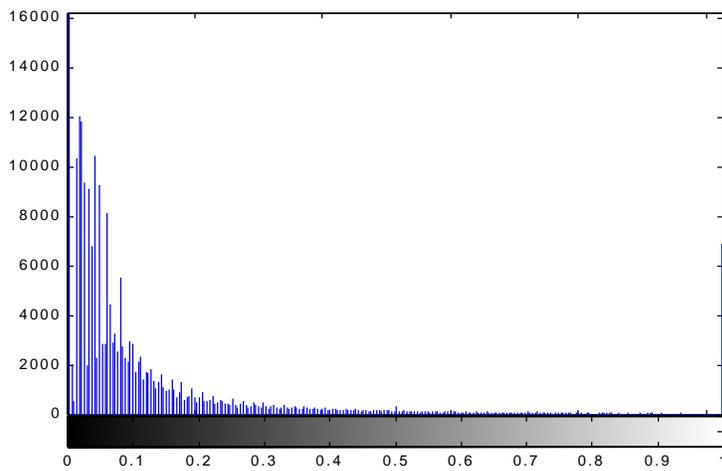
Los resultados que se han obtenido al realizar el histograma a la cabeza de la 5ª persona antes y después del filtrado son los siguientes:



HISTOGRAMA DE LA 5ª CABEZA SIN FILTRAR

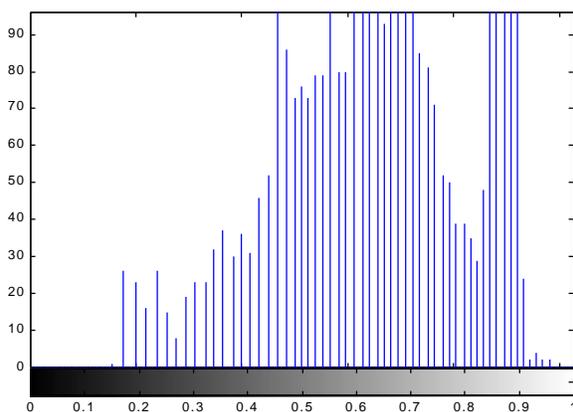


HISTOGRAMA DE LA 5ª CABEZA DESPUÉS F. PREWITT

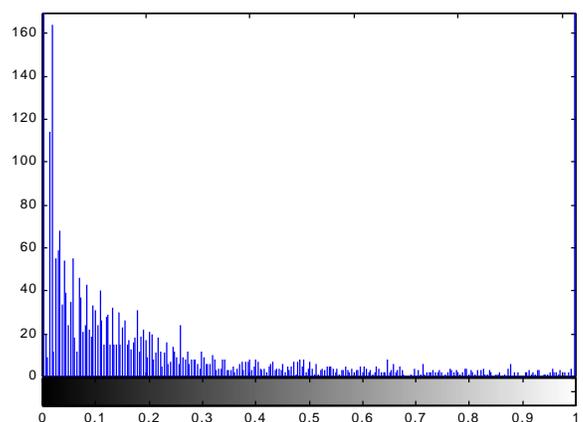


HISTOGRAMA 5ª
CABEZA
DESPUÉS DE
APLICAR
F. SOBEL

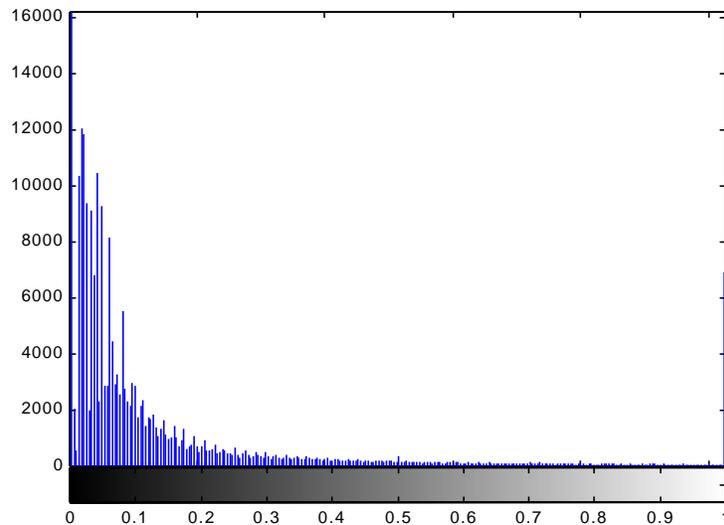
Los resultados que se han obtenido al realizar el histograma a la 6ª cabeza de la imagen Gatlin antes y después del filtrado son los siguientes:



HISTOGRAMA
DE LA 6ª CABEZA
SIN FILTRAR



HISTOGRAMA
DE LA 6ª CABEZA
DESPUÉS F.PREWITT



HISTOGRAMA
DE LA 6ª CABEZA
DESPUÉS F.

Después de todos estos resultados obtenidos observamos una serie de características comunes :

- No se aprecian diferencias notables en el histograma de una cabeza respecto a la otra, es decir, presentan similares características.
- Hay una diferencia significativa en el histograma de una imagen y en el de la imagen transformada al aplicarle un filtro de Prewitt o de Sobel.
- Tampoco se aprecian diferencias en el histograma de una cara y otra cuando previamente a la cara se le ha aplicado una transformación de Prewitt o de Sobel.

Con este estudio realizado pasamos a hacer un análisis pormenorizado de los rostros de cada una de las seis personas con la intención de encontrar que las mismas zonas de la cara de distintas personas presentan rasgos similares. Con esta motivación presento a continuación cada uno de los seis rostros con su análisis por valores correspondiente.

Si conjugamos los resultados obtenidos con las técnicas de umbralización con los filtros de Prewitt se observa que cuando el filtro Prewitt produce una buena detección del contorno perimetral de la cara se produce un ajuste muy fino del contorno si utilizamos un umbral de valor unitario. Hemos ido probando distintos valor de umbral pero el mejor resultado se obtiene para este valor. No obstante hay que resaltar que cuando el contorno no está bien definido los resultados no son buenos.

Los resultados mejores se obtienen para la quinta y la sexta persona empezando por la izquierda arrojando los siguientes resultados:



IMAGEN 5^a
PERSONA
FILTRADA CON
PREWITT

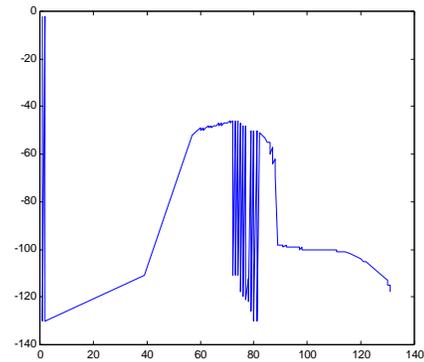


IMAGEN DESPUÉS
DE
PREWITT+ UMBRAL

Para la sexta persona se han arrojado los siguientes resultados:



IMAGEN 6^a
PERSONA

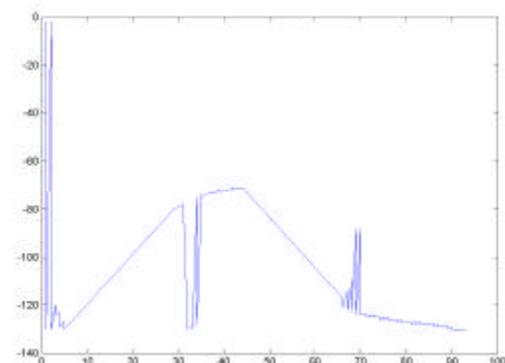
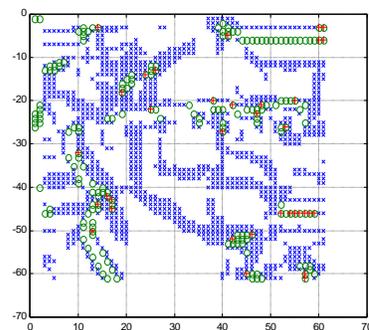


IMAGEN TRAS PREWITT
Y UMBRALIZACIÓN

Aunque este resultado elimina gran parte del ruido en cuanto a señal perimetral indeseada y el resultado es muy bueno para la quinta

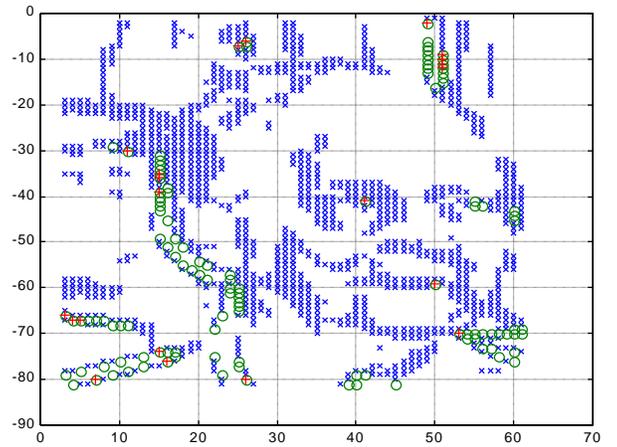
y sexta cabeza de la imagen Gatlin, para el resto de las cabezas el resultado no es tan bueno. Para este caso bastaría con bajar el umbral unitario a un umbral de 0.5 para obtener resultados notables. Hay que ser consciente que al aplicar filtro de Prewitt a la imagen los ojos toman un valor que oscila entre 0.5 y 0.53 y si quiero obtener una presencia de ellos en la imagen transformada debo poner el umbral por debajo de este valor. Ahondando en lo citado anteriormente voy a realizar un análisis por valores de las distintas zonas de la cara con la motividad de encontrar similitudes.

La cara de la primera persona arroja los siguientes resultados:

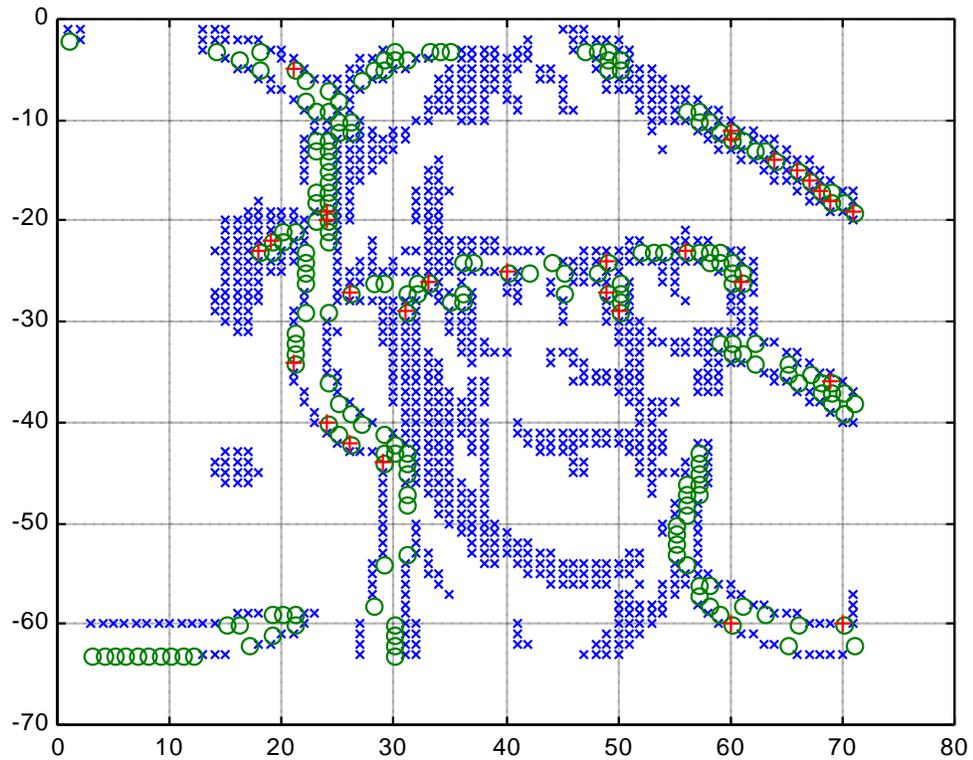


Como se puede contemplar del resultado obtenido los ojos se corresponden con la zona donde están dibujadas las cruces rojas. Si esto sólo se produjera en una cara el tema carecería de importancia pero al ser una tónica general que se repite en los demás casos será un punto a valorar y a tener en cuenta. A continuación voy a representar los resultados de las demás caras con el objetivo de ratificar el resultado.

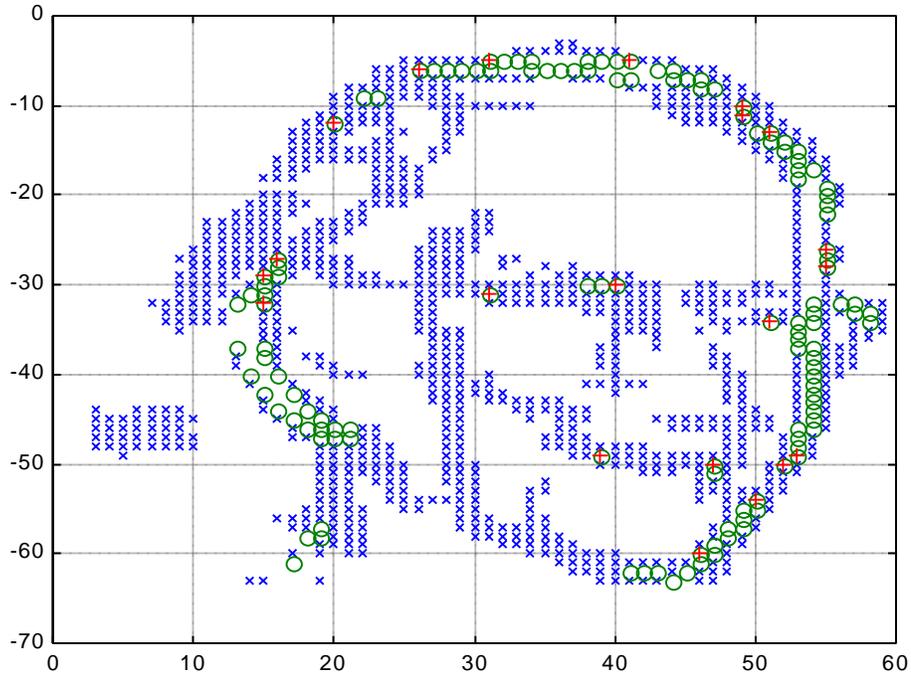
La cara de la segunda persona arroja los siguientes resultados:



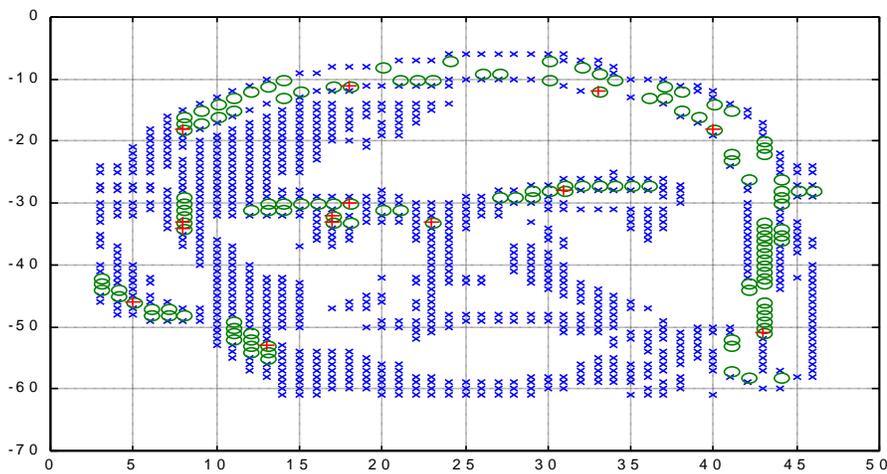
La cara de la tercera persona arroja los siguientes resultados:



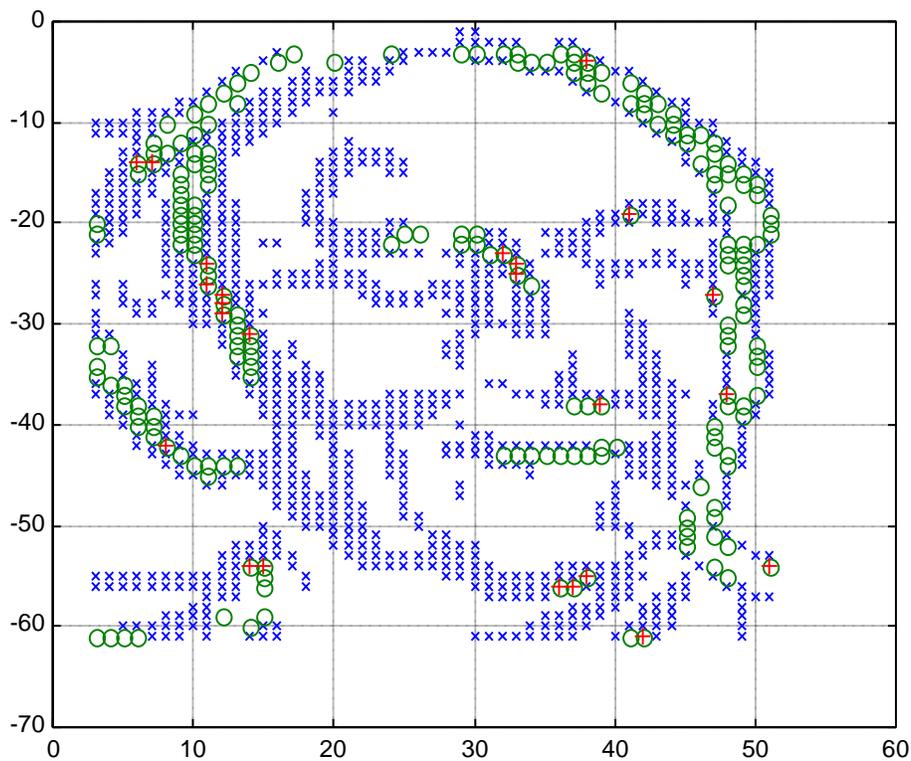
En cuanto a la cara de la cuarta persona se obtienen los siguientes resultados:



La cara de la quinta persona arroja los siguientes resultados:



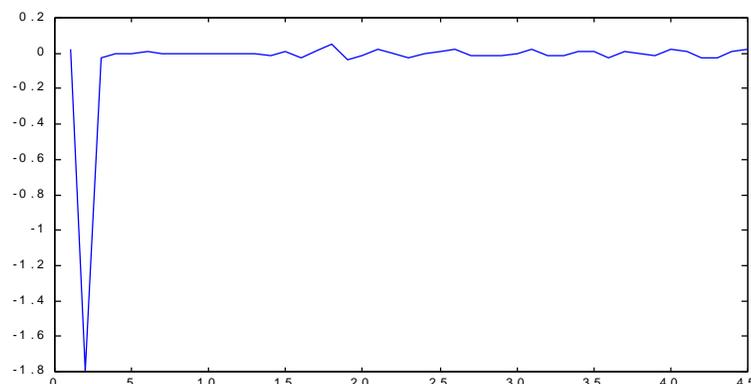
La cara de la sexta persona arroja los siguientes resultados:



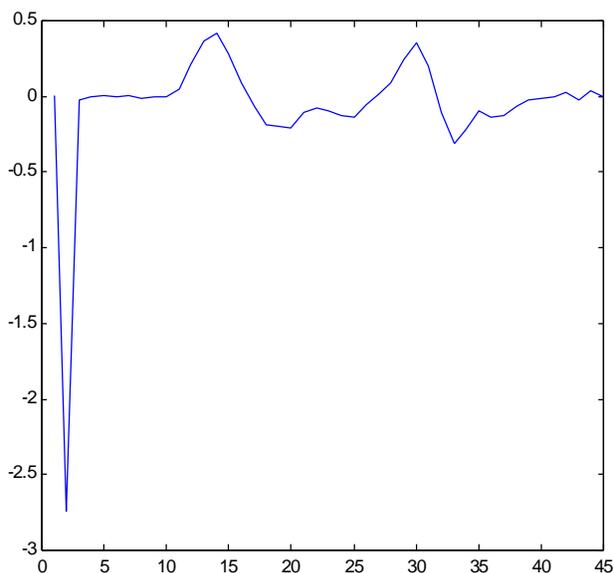
Con este análisis de la imagen filtrada por valores hemos conseguido detectar donde se encuentran los ojos. Esta técnica en principio tan visual la completaremos a continuación. Ayudado por los resultados obtenidos nos decidimos por realizar un estudio del contorno de la cabeza fila por fila y columna por columna. En lo que respecta al análisis de las filas de la imagen lo que hemos hecho es realizar en primer lugar un filtrado de Prewitt sobre la imagen original con el objetivo de disminuir el grado de información que hay en la

imagen original para que sucesivos procesamientos sobre la misma se realicen con un menor coste computacional. Acto seguido de la imagen filtrada me quedo con una fila aislada y sobre la misma realizo una derivada para ver donde se producen las variaciones en la señal porque dichas variaciones serán indicadores de que ahí se ha producido un salto brusco y por consiguiente se ha detectado un contorno. Este análisis se ha hecho para todas las filas de la imagen. Con ello a priori intentamos no sólo detectar donde se encuentran los contornos perimetrales de la cara sino también la ubicación de los ojos ya que la zona donde se encuentran los mismo debe presentar un mayor número de montañas y valles ya que es donde se produce más cambios de intensidad en los contornos. Este estudio se ha realizado para todas y cada una de las personas que aparecen en la imagen Gatlin. Sin embargo como para cada persona he realizado del orden de 61x46 gráficas la inclusión de todas ellas en la memoria resultaría engorroso. No obstante, con objeto de que se puedan hacer una idea de los resultados obtenidos mostraré los resultados más representativos conseguidos sobre la quinta cabeza empezando por la izquierda de la imagen Gatlin.

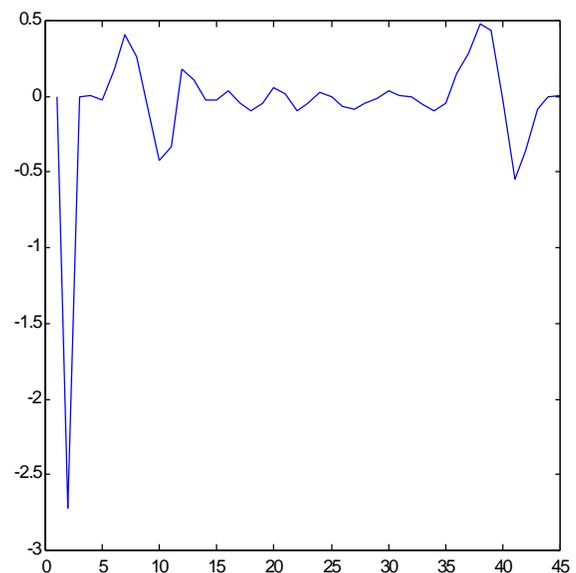
En el análisis de la primera fila como no se aprecian cambios en la imagen es lógico que la gráfica resultante del proceso derivativo sea la siguiente:



En la filas 6 y 7 de la imagen que es cuando se empieza a coger la parte superior de la cabeza se manifiesta un cambio en la derivada en los píxeles 25-30 del eje de abscisas, que manifiesta el ensanchamiento que va sufriendo la cabeza a medida que se va bajando desde la parte superior del cráneo. La representación gráfica que atestigua este resultado es el siguiente:



FILA 10



FILA 16

En lo que respecta a la detección de contornos en la zona donde se ubica los ojos, se aprecia que por nuestra constitución física se deben encontrar por encima de la mitad superior de la imagen. Si apoyados por esta certeza física nos apoyamos en los gráficos obtenidos se observa que los ojos se encuentran ubicados en las filas

que van desde 31 a la 37. No obstante, como este margen de error es considerablemente grande lo que hemos realizado es una función que detecte el número de pasos por cero de la gráfica resultante de la derivada de cada fila de la imagen. Con esto obtenemos cual es la fila de la imagen que presenta un mayor número de pasos por cero en la imagen, que se corresponderá con la zona donde se encuentran los ojos. Para ello he realizado un bucle que vaya recorriendo cada elemento columna de una fila determinada y mediante la multiplicación del valor de dos píxeles consecutivos sé si su valor es negativo que se ha producido un paso por cero y lo iré acumulando en un contador que lo he llamado "contadorpasosporcero". Sin embargo, como tras la ejecución de la función anterior van a salir un número considerable de valores y para facilitar su estudio comparativo que es el que al final nos interesa hemos realizado otra función que mostrará los resultados anteriores pero dividido por el máximo valor de todos los valores de la imagen. Así, por ejemplo, para la quinta persona del Gatlin los valores arrojados por la primera de las funciones descritas son los siguientes:

I1=I(40:100,445:490)

contadorpasosporcero =

Columns 1 through 12

8 14 19 15 13 4 4 4 9 7 11 14

Columns 13 through 24

17 12 13 14 9 12 13 16 14 12 12 11

Columns 25 through 36

8 8 13 14 10 15 19 20 16 14 16 15

Columns 37 through 48

22 18 17 16 17 14 15 14 16 14 14 17

Columns 49 through 60

16 13 16 17 16 15 15 14 17 13 10 15

Column 61

12

Los resultados obtenidos al realizar el estudio comparativo fueron los siguientes:

contadorpasosporcero =

Columns 1 through 7

0.3636 0.6364 0.8636 0.6818 0.5909 0.1818 0.1818

Columns 8 through 14

0.1818 0.4091 0.3182 0.5000 0.6364 0.7727 0.5455

Columns 15 through 21

0.5909 0.6364 0.4091 0.5455 0.5909 0.7273 0.6364

Columns 22 through 28

0.5455 0.5455 0.5000 0.3636 0.3636 0.5909 0.6364

Columns 29 through 35

0.4545 0.6818 0.8636 0.9091 0.7273 0.6364 0.7273

Columns 36 through 42

0.6818 1.0000 0.8182 0.7727 0.7273 0.7727 0.6364

Columns 43 through 49

0.6818 0.6364 0.7273 0.6364 0.6364 0.7727 0.7273

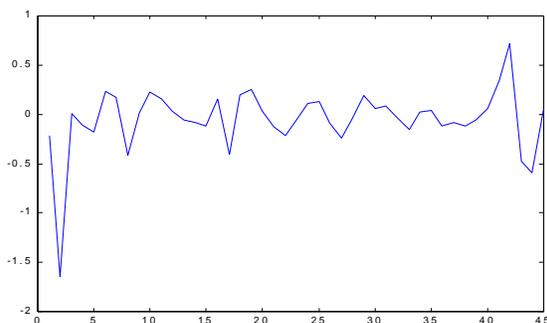
Columns 50 through 56

0.5909 0.7273 0.7727 0.7273 0.6818 0.6818 0.6364

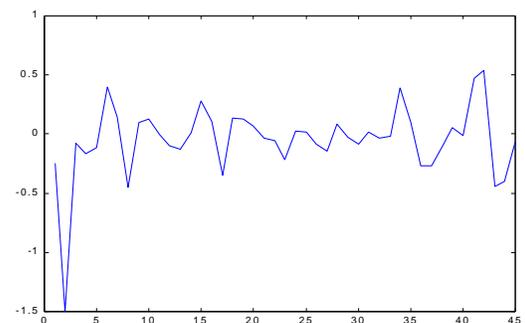
Columns 57 through 61

0.7727 0.5909 0.4545 0.6818 0.5455

Con este estudio se reduce el amplio margen de error que teníamos entre las filas que iban desde la 31 a la 37. Ahora sabemos concretamente que los ojos por su situación dentro de la cara y por los resultados obtenidos se encuentran en las filas 31 y 32. También hay que considerar que se encuentran un valor elevado en las filas 37 y 38 que se corresponde con la zona donde se ubica la nariz. Las gráficas de la zona donde se encuentran los ojos son las siguientes:



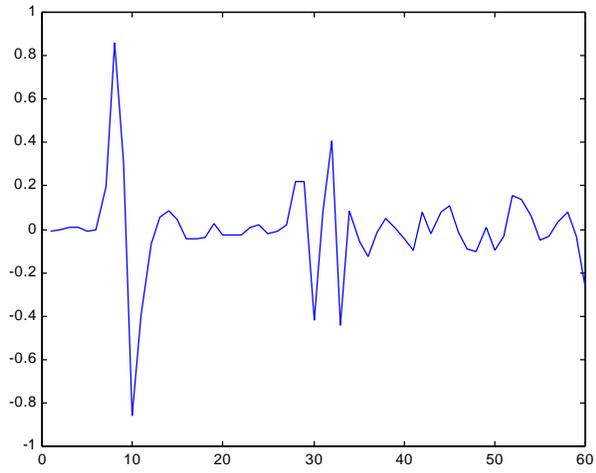
FILA 31



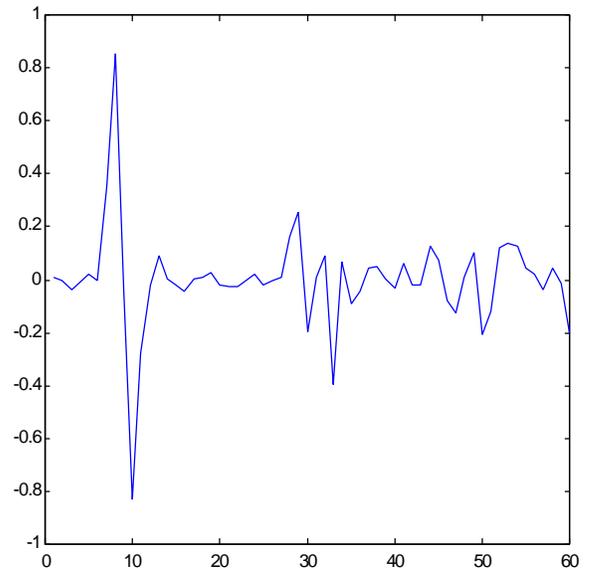
FILA 32

En cambio la zona de la boca se sitúa entre las filas 51 a 53 porque no sólo presenta un mayor valor relativo el contador de pasos por cero sino que además ese valor se encuentra entorno a la cuarta parte de las filas empezando por la primera fila que se sitúa en la parte superior de la imagen.

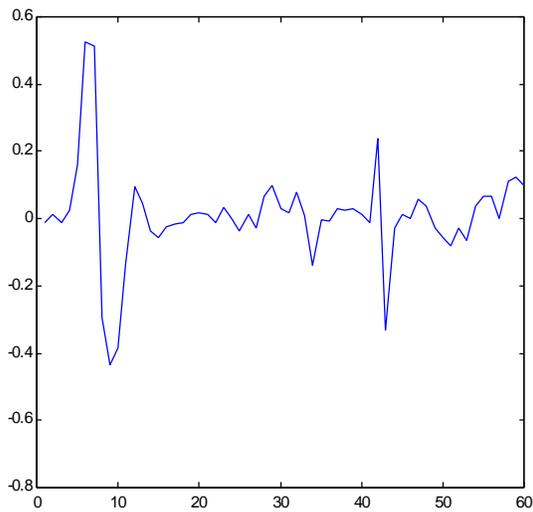
A continuación debemos correlacionar este resultado con el que se obtiene por columnas para así poder determinar la zona donde se encuentra los ojos, boca y el contorno. En el análisis por columnas se observa un valor alto en las columnas 18 y 19 respecto a las demás, pero si realizamos un análisis comparativo de cada columna respecto a la columna que tenga el máximo observo valores altos en las columnas 4,18,19,24,25,38,39. Como son 46 columnas los ojos deben estar situados a izquierda y derecha de la línea divisoria de los mismos que se sitúa en 23. Bajo estos supuestos que tienen un fundamento fuerte en nuestra estructura física establezco que con los resultados obtenidos el ojo derecho se sitúa en las columnas 18,19 y el ojo izquierdo en las columnas 24,25. Las gráficas que muestran estos resultados son las siguientes:



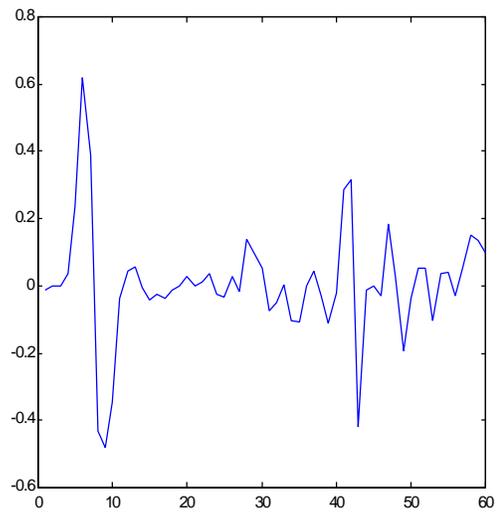
FILA 18



FILA 19



FILA 24



FILA 25

Si los resultados arrojados del análisis por columnas los correlacionamos con el análisis por filas se obtienen los siguientes resultados:

- El ojo derecho se encuentra situado en las filas 31 y 32 y en las columnas 18 y 19.
- El ojo izquierdo se encuentra situado entre las filas 31 y 32 y entre las columnas 24 y 25.
- La nariz se encuentra situada en la columna 23 y en la fila 37.
- La boca se encuentra situada entre las filas 51 y 53.

Por tanto como se puede apreciar, gracias al conjunto de varias técnicas se obtiene un resultado que determina la zona donde se ubican los accidentes de la cara.

CUARTA PARTE. CONCLUSIONES

Como consecuencia de la gran cantidad de información que se encuentra en la imagen y partiendo de la necesidad de reducir esta información para poder determinar accidentes de la cara hemos utilizado diferentes técnicas.

En primer lugar realizamos el histograma sobre la imagen original y con ello obtuvimos la frecuencia con la que aparecían los distintos valores de la imagen. Seguidamente decidimos a la imagen original ponerle un umbral pero esto tampoco nos servía para detectar los contornos porque lo que hacíamos era perder información ya que los píxeles por debajo de un umbral los hacia cero, oscureciendo la imagen. Como esto no era suficiente nos adentramos con un filtro de gradiente. Dentro de los filtros de gradiente hemos utilizado el filtro de Roberts para detectar los contornos diagonales en ángulos que forman 45° y 135° . El resultado que se ha obtenido con la aplicación de este filtro ha resultado ser bastante pobre, ya que se extraía poco contenido de la imagen original. Por si fuera poco, los resultados se obtenían evaluando pocos píxeles del entorno de vecindad, con lo cual cualquier elemento perturbador que hubiera cerca de las personas podía afectar a la imagen. Por ello, continuamos adelante, y si bien antes nos fuimos por un filtro que detectaba los contornos diagonales, ahora nos vamos por uno que detectaba los contornos tanto verticales como horizontales. Para ello realizamos una función que hiciera un procesado sobre la imagen aplicándole un filtro de Sobel. Con el filtro de Sobel realizamos primero un procesado en el que detectábamos

los contornos horizontales, apareciendo estos muy bien definidos sobretudo en comparación con el filtro de Roberts. Después realizamos los contornos verticales obteniendo un resultado tan bueno como el nombrado en el caso de los contornos horizontales. Esto sin embargo no se podía dejar así y decidimos extraer lo mejor de los dos resultados antes obtenidos (las dos imágenes, en una se detectaban los contornos horizontales y en otra los verticales). Para ello hemos ido recorriendo esas dos imágenes y colocando en una matriz el valor más luminoso de las dos imágenes anteriores. El realce de las distintas partes de la imagen es bueno, pero si nos fijamos bien en la cabeza el resultado sólo es muy bueno en las personas situadas en la posición 5ª y 6ª.

A pesar de que el resultado anterior era bueno, decidí utilizar el filtro de Prewitt con la intención de ver los resultados que se obtenían de él. Practicamos la misma forma de operar de antes, construyendo una función que detecte en primer lugar los contornos horizontales, posteriormente los verticales y en última instancia construimos una matriz que obtenga lo mejor de los dos resultados anteriores. El resultado que se ha obtenido tras aplicar el filtro de Prewitt no presenta notables diferencias respecto al obtenido con el filtro de Sobel.

Haciendo un análisis conjunto de lo que hemos obtenido hasta el momento hay que resaltar que se ha obtenido un mejor resultado con los filtros de Sobel y Prewitt que con los filtros de gradiente y Roberts.

En este momento nos surgió la idea, de que al ser la cabeza en elemento cuasi rectangular o elíptico al convolucionarlo la imagen de cuerpo entero de una persona con un rectángulo se obtendría píxeles con una mayor luminosidad en la zona de la cara que es donde coincide el rectángulo con la cara. Pero nada más lejos de la realidad. Lo único que conseguimos con esta técnica era realzar todos los contornos del cuerpo y la línea de contorno que antes era fina ahora era gruesa.

Después de haber acometido los filtrados anteriores mediante la realización de funciones propias decidimos utilizar una función de la librería de Matlab que se llama "EDGE" a la cual le aplicamos distintos parámetros para obtener distintos tipos de filtrado. En primer lugar, le aplicamos el operador de Roberts, obteniéndose un resultado bastante pobre. Seguidamente, como parámetros usamos Sobel y Prewitt obteniéndose resultados tan buenos como los vistos con anterioridad y finalmente le aplicamos como parámetro "LOG" y "ZEROCROSS" obteniéndose unos resultados mejores que los anteriores ya que se cierra un poco más los contornos antes obtenidos con Sobel y Prewitt.

Por último, decidimos aplicar el filtro de Frei- Chen pero lo resultados fueron peores que con los filtros de Sobel y Prewitt.

Después de todo el estudio realizado, decidimos encontrar zonas características de la cara con el objetivo de poder distinguir a una persona de otra. Como las características del histograma son bien distintas en la imagen original y en la imagen filtrada decidimos

exponer los resultados que se obtenían en uno y otro caso. El histograma podría ser una buena técnica de identificación de la persona, con la única salvedad que el histograma debería aplicarse sobre la imagen sin filtrar ya que el histograma que se obtiene sobre la imagen filtrada es muy similar para todas las caras.

A continuación pasamos a hacer un análisis pormenorizado por valores de cada una de las seis personas que aparecen en la imagen "Gatlin", con la motivación de encontrar características comunes en las caras. Para ello, utilizamos un filtro de Prewitt y representamos con distintos colores determinados intervalos de intensidad de los píxeles, con el objeto de ver que la misma zona de la cara presenta los mismos valores en una persona y en otra. Posteriormente, con el conocimiento de los valores de cada una de las caras decidimos representar el contorno de la misma después de aplicar el filtro de Prewitt. El resultado fue bueno para la quinta y sexta imagen, donde el contorno perimetral de la cabeza está bien marcado no siéndolo para el resto. Este notable resultado se obtuvo para un umbral de valor unitario que nos permitía obtener un contorno muy limpio de la imagen original. De este análisis por valores de la cara hay que mencionar que los ojos se encuentran con unos valores de intensidad entre 0.5 y 0.53. Si este hecho sólo se produjese en una de las caras el hecho carecería de importancia, pero al ser una tónica que se repite en el resto de las caras este hecho adquiere importancia y permitiría determinar en que zona se encuentran los ojos.

Ayudado por los resultados obtenidos nos decidimos por realizar un estudio del contorno de la cabeza fila por fila y columna por

columna. Para disminuir la información de la imagen original utilizamos un filtro de Prewitt y sobre ella hemos cogido fila a fila y sobre cada fila, para detectar el contorno le aplicamos la derivada, ya que ésta nos permitirá realzar cambios bruscos que indican que ahí hay un contorno. Estos resultados se han obtenido para todas las filas pudiéndose detectar como se ensancha la cabeza debido a que se empieza a procesar por la parte superior de la misma. En cuanto a las filas, nuestra idea de partida era que la zona donde se ubican los ojos debe presentar una mayor alternancia en cuanto a los tonos de intensidad de los píxeles. Esto se podía apreciar visualmente viendo fila a fila, pero para facilitar la labor, implementamos una función que calculase el número de pasos por cero de cada fila de la imagen. Con esto se determina la fila que presenta un mayor número de pasos por cero siendo esta la que corresponde con la zona donde se ubican los ojos. Este hecho debe ser corroborado por la certeza física de que los ojos se sitúan por encima de la mitad superior de la cabeza.

Una vez realizado un resumen de los pasos que hemos seguido para la ejecución del proyecto es necesario esbozar las fortalezas.

Las fortalezas de los métodos aplicados reside:

- La sencillez de los métodos propuestos.
- La rapidez con la que se obtienen los resultados, que permite ejecutar los mismo en tiempo real.

En cuanto a las líneas de investigación que se deben abordar para continuar en un proceso de mejora continua son las siguientes:

- Plantearse otros métodos para abordar el problema de detección de contornos con los códigos de cadena, transformada Hough. Estos métodos pueden servir de apoyo para aquellos casos donde los métodos anteriores no funcionaban del todo bien.
- Realizar dichas funciones en lenguaje C con objeto de optimizar el código. No hace falta que toda la función esté realizada en lenguaje C sino sólo la parte de aquella función que es repetitiva, es decir, los bucles.

ANEXO

CONTADOR PASOS POR CERO POR FILAS

El contador de los pasos por cero de todas las filas de la imagen de la primera persona arroja los siguientes resultados:

I1=I(60:130,50:110)

contadorpasosporcero =

Columns 1 through 12

16 16 23 15 9 9 16 17 14 13 15 11

Columns 13 through 24

13 14 22 14 17 26 20 20 20 17 20 20

Columns 25 through 36

16 16 19 20 22 18 20 15 18 19 26 25

Columns 37 through 48

19 23 19 22 14 14 14 26 20 19 17 18

Columns 49 through 60

15 15 22 22 16 15 13 14 19 17 14 15

Columns 61 through 71

9 15 16 14 18 17 10 15 13 13 12

De los resultados obtenidos hay que mencionar que aparecen valores altos en las filas 18, 35, 36, 44, 51 y 52. A continuación como estamos considerando sólo las filas dividimos la cara en dos mitades iguales. En la mitad superior se situarán los ojos y en la mitad inferior la boca. Si la mitad superior de la cabeza la dividimos en dos partes los ojos se situarán en la mitad inferior. Con los resultados obtenidos y apoyándonos en la en estas hipótesis tenemos que los ojos se sitúan en las filas 35 y 36 mientras que la boca se sitúa en las filas 51 y 52. Posteriormente este resultado lo cruzaré con el obtenido para las columnas para determinar la posición exacta de los accidentes de la cara.

El contador de los pasos por cero de la segunda persona origina el siguiente resultado:

$I1=I(60:130,150:210)$

contadorpasosporcero =

Columns 1 through 12

21 20 18 14 20 23 24 22 24 19 22 20

Columns 13 through 24

20 26 24 24 17 22 21 24 21 20 19 22

Columns 25 through 36

19 22 19 21 17 23 22 23 20 20 22 21

Columns 37 through 48

22 24 22 20 22 20 24 26 18 19 17 19

Columns 49 through 60

21 19 19 19 20 18 25 19 20 21 21 20

Columns 61 through 71

18 24 20 17 18 15 10 16 16 13 12

Siguiendo el mismo procedimiento descrito con anterioridad podemos determinar que los ojos se encuentran en las filas 30, 31 y 32 mientras que la boca se encuentra en la fila 55 por presentar el contador de pasos por cero un valor alto y situarse dentro de la mitad de la mitad inferior de la cara.

El contador de los pasos por cero de la tercera persona arroja los siguientes resultados:

$I1=I(55:120,225:280)$

contadordpasosporcero =

Columns 1 through 12

13 24 17 14 11 11 11 15 14 13 17 13

Columns 13 through 24

11 15 14 16 19 17 11 13 17 16 15 15

Columns 25 through 36

17 12 16 20 21 18 21 16 17 12 16 16

Columns 37 through 48

14 16 17 17 11 18 12 17 19 15 12 18

Columns 49 through 60

19 12 15 15 17 12 11 14 15 14 16 16

Columns 61 through 66

17 15 10 15 11 15

Gracias a los resultados arrojados se ha obtenido que los ojos se encuentran entre las filas 28, 29, 30 y 31 mientras que la boca se sitúa en las filas 48 y 49.

El contador pasos por cero de la cuarta persona arroja los siguientes resultados:

$I1=I(60:130,340:400)$

contadorpasosporcero =

Columns 1 through 12

10 21 11 10 12 12 25 17 19 16 16 18

Columns 13 through 24

18 16 18 17 21 22 18 20 20 14 17 23

Columns 25 through 36

20 26 21 14 15 25 23 21 23 25 27 24

Columns 37 through 48

21 19 16 18 23 15 22 15 14 18 21 19

Columns 49 through 60

23 21 17 14 13 22 21 14 26 24 19 15

Columns 61 through 71

24 17 22 18 15 12 12 16 11 17 22

Según los resultados obtenidos los ojos se sitúan en las filas 26, 30, 34 ó 35. Si embargo como la zona donde se ubican los ojos no es puntual y entre las filas que van de la 30 a la 35 hay valores altos de los contadores podemos decir que los ojos se encuentran a partir de la fila 30. La boca se sitúa en la fila 49, 50 ya que la imagen recoge para esta fila un valor alto de contador. No podemos considerar el contador de la fila 57 porque este punto se corresponde con la barbilla de la cara.

El contador pasos por cero de la quinta persona produce los siguientes resultados:

I1=I(40:100,445:490)

contadorpasosporcero =

Columns 1 through 12

8 14 19 15 13 4 4 4 9 7 11 14

Columns 13 through 24

17 12 13 14 9 12 13 16 14 12 12 11

Columns 25 through 36

8 8 13 14 10 15 19 20 16 14 16 15

Columns 37 through 48

22 18 17 16 17 14 15 14 16 14 14 17

Columns 49 through 60

16 13 16 17 16 15 15 14 17 13 10 15

Column 61

12

Los resultados de la quinta persona ya han sido comentados con anterioridad.

El contador de pasos por cero de la sexta persona arroja los siguientes resultados:

I1=I(60:130,550:600)

contadorpasosporcero =

Columns 1 through 12

12 18 9 13 14 13 12 19 15 12 13 12

Columns 13 through 24

10 13 13 10 11 15 11 14 12 13 15 10

Columns 25 through 36

14 13 14 16 14 20 9 12 16 16 14 20

Columns 37 through 48

18 18 17 16 17 21 15 22 19 20 16 13

Columns 49 through 60

17 15 15 17 16 16 16 18 20 20 17 16

Columns 61 through 71

13 14 14 12 18 13 15 17 16 13 10

De los resultados arrojados y dividiendo el número de filas por la mitad tenemos que los ojos se encuentran en la fila 30 mientras que la boca en la fila 57, 58.

CONTADOR PASOS POR CERO POR COLUMNA

El contador pasos por cero al pasarlo sobre la primera persona arroja los siguientes resultados:

contadorpasosporcerocol =

Columns 1 through 12

10 13 21 25 19 21 19 23 21 17 19 23

Columns 13 through 24

26 25 23 19 23 23 24 19 19 20 24 26

Columns 25 through 36

22 24 24 17 12 20 20 18 20 21 23 24

Columns 37 through 48

22 25 20 17 16 19 24 23 24 19 18 22

Columns 49 through 60

24 25 18 24 22 27 26 17 16 18 21 21

Column 61

20

En esta imagen el hecho que la persona lleve gafas acarrea problemas en la determinación de la columna en la que se encuentra el ojo ya que podemos decir que el ojo izquierdo se encuentra situado entre las columnas [23-27] mientras que el ojo derecho se encuentra entre las columnas [35-38]. La boca también es difícil de determinar pero sin embargo podemos hacer notar que la boca se debe encontrar entre las columnas 23-38 que son los extremos que determinan donde se encuentran los ojos. No obstante, el resultado que se ha obtenido no es bueno porque la cara de esta persona aparece ladeada y por tanto al dividir el número de columnas por la mitad y decir que en la mitad izquierda encontramos el ojo izquierdo y en la mitad derecha el ojo derecho es falsa. No obstante, visualizando la cara de la primera persona en el análisis por valores podemos comprobar que los ojos se sitúan en la columna 38 y 50 respectivamente presentando en dicha posición sus contadores respectivos un valor de 25.

El contador pasos por cero al pasarlo sobre la segunda persona arroja los siguientes resultados:

contadorpasosporcerocol =

Columns 1 through 12

7 8 19 25 15 21 20 18 16 16 18 21

Columns 13 through 24

13 19 21 14 13 20 21 20 16 17 15 16

Columns 25 through 36

17 18 22 21 18 24 27 21 26 19 20 23

Columns 37 through 48

24 25 20 16 13 18 20 27 24 21 25 22

Columns 49 through 60

24 19 19 20 19 20 21 11 9 11 13 14

Column 61

12

Después de obtener estos resultados se concluye que el ojo izquierdo se encuentra situado en la columna 31 y el ojo derecho en la columna 44. En la media aritmética de 31 y 44 $((31+44)/2)$ se encuentra situada la nariz.

El contador pasos por cero al pasarlo sobre la tercera persona arroja los siguientes resultados:

contadorpasosporcerocol =

Columns 1 through 12

4 6 13 21 19 17 19 14 13 11 9 13

Columns 13 through 24

15 13 10 12 17 15 18 20 17 25 16 18

Columns 25 through 36

18 18 15 15 12 14 17 20 17 20 24 19

Columns 37 through 48

19 20 20 23 21 19 21 21 19 17 19 21

Columns 49 through 56

24 22 19 18 17 15 13 17

De los resultados arrojados se concluye que el ojo izquierdo se encuentra en la columna 35 mientras que el ojo derecho se encuentra en la columna 49. De esta imagen hay que resaltar que las primeras 15 columnas no pertenecen a la cara de la persona en cuestión, así que lo que se ha hecho es proceder a su eliminación a la hora de dividir la imagen en dos partes iguales.

El contador pasos por cero al pasarlo sobre la cuarta persona arroja los siguientes resultados:

contadorpasosporcerocol =

Columns 1 through 12

7 4 15 17 24 24 20 17 22 25 27 24

Columns 13 through 24

19 17 18 12 18 16 21 20 24 27 21 28

Columns 25 through 36

24 25 20 22 22 19 21 24 22 22 27 22

Columns 37 through 48

14 16 18 14 22 17 14 16 14 16 25 11

Columns 49 through 60

12 13 20 11 18 20 9 7 10 12 13 10

Column 61

20

Esta imagen presenta el inconveniente de tener la cara ladeada. No obstante a pesar de esta adversidad, de los resultados arrojados establecemos que los ojos se encuentran en las columnas 24 y 35.

El contador pasos por cero al pasarlo sobre la quinta persona arroja los siguientes resultados:

contadorpasosporcerocol =

Columns 1 through 12

6 4 19 21 19 15 16 13 11 17 12 14

Columns 13 through 24

12 13 12 20 13 26 24 18 15 20 14 20

Columns 25 through 36

20 16 24 16 17 20 19 18 19 17 16 19

Columns 37 through 46

18 20 22 19 11 13 7 9 16 13

El ojo derecho de la quinta persona de la imagen 'gatlin' se encuentra situado entre las columnas 18 y 19, mientras que el ojo izquierdo lo hace en la columna 27.

El contador pasos por cero al pasarlo sobre la sexta persona arroja los siguientes resultados:

contadorpasosporcerocol =

Columns 1 through 12

8 12 22 20 18 20 23 20 23 22 17 14

Columns 13 through 24

24 17 19 24 18 21 21 25 23 25 24 19

Columns 25 through 36

17 24 20 22 25 19 19 19 23 24 26 27

Columns 37 through 48

20 27 22 28 24 21 22 19 19 14 15 13

Columns 49 through 51

12 15 15

El ojo derecho de la sexta persona de la imagen 'gatlin' se ubica entre las columnas 20, 21, 22, 23 presentando un mayor valor del contador en la columna 22. El ojo izquierdo se encuentra situado entre las columnas [35-40], presentando su mayor valor en la columna 40 que es donde se sitúa el ojo.

CONTADOR PASOS POR CERO COLUMNAS EN BLANCO Y NEGRO

El contador pasos por cero columna de la primera persona arroja los siguientes resultados:

contadordpasosporcerocol =

Columns 1 through 12

0 0 1 0 1 1 2 3 2 1 1 1

Columns 13 through 24

1 0 1 1 1 2 3 0 2 0 1 0

Columns 25 through 36

0 0 0 0 1 1 0 0 0 2 1 0

Columns 37 through 48

0 0 4 1 2 0 1 2 1 4 2 3

Columns 49 through 60

1 1 2 1 1 2 2 0 0 3 2 2

Column 61

0

El contador pasos por cero columna de la segunda persona arroja los siguientes resultados:

contadorpasosporcerocol =

Columns 1 through 12

0 0 1 2 0 0 0 0 1 1 0 1

Columns 13 through 24

1 0 0 0 0 0 0 0 0 0 0 0

Columns 25 through 36

0 2 0 0 1 2 2 3 2 2 2 2

Columns 37 through 48

2 2 3 3 2 1 1 0 0 0 1 2

Columns 49 through 60

2 0 0 0 0 1 0 0 0 1 0 0

Column 61

1

El contador pasos por cero columna de la tercera persona arroja los siguientes resultados:

contadorpasosporcerocol =

Columns 1 through 12

0 0 0 0 0 0 0 0 0 0 0 0

Columns 13 through 24

0 0 0 0 0 0 0 2 0 0 0 0

Columns 25 through 36

1 0 0 0 1 1 1 1 1 1 0 0

Columns 37 through 48

0 1 1 1 1 2 2 3 3 2 1 1

Columns 49 through 56

1 3 2 1 1 0 0 0

El contador pasos por cero columna de la cuarta persona arroja los siguientes resultados:

contadorpasosporcerocol =

Columns 1 through 12

0 0 0 0 0 0 2 0 0 0 0 0

Columns 13 through 24

1 0 0 0 0 0 1 1 1 1 0 0

Columns 25 through 36

1 1 1 1 0 2 1 0 2 3 1 2

Columns 37 through 48

0 1 0 1 0 0 2 0 0 0 0 0

Columns 49 through 60

0 0 0 0 0 0 0 0 0 0 0 0

Column 61

0

El contador pasos por cero columna de la quinta persona arroja los siguientes resultados:

contadordpasosporcerocol =

Columns 1 through 12

0 0 0 0 1 1 0 0 0 0 0 0

Columns 13 through 24

0 0 1 2 0 2 2 0 1 0 0 1

Columns 25 through 36

1 0 0 0 0 0 0 2 2 0 0 0

Columns 37 through 46

1 0 0 1 0 1 0 0 0 0

El contador pasos por cero columna de la sexta persona arroja los siguientes resultados:

contadordpasosporcerocol =

Columns 1 through 12

0 0 0 0 0 1 0 0 1 3 1 1

Columns 13 through 24

1 2 3 4 2 0 1 0 0 1 1 1

Columns 25 through 36

1 1 1 0 0 2 1 0 1 2 2 3

Columns 37 through 48

0 4 3 2 3 3 2 2 0 0 0 0

Columns 49 through 51

0 0 0

CONTADOR PASOS POR CERO/ MAX(CONTADOR PASOS POR CERO)

El contador pasos por cero en tanto por uno de la primera persona arroja los siguientes resultados:

contadorpasosporcero =

Columns 1 through 7

0.6154 0.6154 0.8846 0.5769 0.3462 0.3462 0.6154

Columns 8 through 14

0.6538 0.5385 0.5000 0.5769 0.4231 0.5000 0.5385

Columns 15 through 21

0.8462 0.5385 0.6538 1.0000 0.7692 0.7692 0.7692

Columns 22 through 28

0.6538 0.7692 0.7692 0.6154 0.6154 0.7308 0.7692

Columns 29 through 35

0.8462 0.6923 0.7692 0.5769 0.6923 0.7308 1.0000

Columns 36 through 42

0.9615 0.7308 0.8846 0.7308 0.8462 0.5385 0.5385

Columns 43 through 49

0.5385 1.0000 0.7692 0.7308 0.6538 0.6923 0.5769

Columns 50 through 56

0.5769 0.8462 0.8462 0.6154 0.5769 0.5000 0.5385

Columns 57 through 63

0.7308 0.6538 0.5385 0.5769 0.3462 0.5769 0.6154

Columns 64 through 70

0.5385 0.6923 0.6538 0.3846 0.5769 0.5000 0.5000

Column 71

0.4615

El contador pasos por cero en tanto por uno de la segunda persona arroja los siguientes resultados:

contadorpasosporcero =

Columns 1 through 7

0.8077 0.7692 0.6923 0.5385 0.7692 0.8846 0.9231

Columns 8 through 14

0.8462 0.9231 0.7308 0.8462 0.7692 0.7692 1.0000

Columns 15 through 21

0.9231 0.9231 0.6538 0.8462 0.8077 0.9231 0.8077

Columns 22 through 28

0.7692 0.7308 0.8462 0.7308 0.8462 0.7308 0.8077

Columns 29 through 35

0.6538 0.8846 0.8462 0.8846 0.7692 0.7692 0.8462

Columns 36 through 42

0.8077 0.8462 0.9231 0.8462 0.7692 0.8462 0.7692

Columns 43 through 49

0.9231 1.0000 0.6923 0.7308 0.6538 0.7308 0.8077

Columns 50 through 56

0.7308 0.7308 0.7308 0.7692 0.6923 0.9615 0.7308

Columns 57 through 63

0.7692 0.8077 0.8077 0.7692 0.6923 0.9231 0.7692

Columns 64 through 70

0.6538 0.6923 0.5769 0.3846 0.6154 0.6154 0.5000

Column 71

0.4615

El contador pasos por cero en tanto por uno de la tercera persona arroja los siguientes resultados:

contadorpasosporcero =

Columns 1 through 7

0.5417 1.0000 0.7083 0.5833 0.4583 0.4583 0.4583

Columns 8 through 14

0.6250 0.5833 0.5417 0.7083 0.5417 0.4583 0.6250

Columns 15 through 21

0.5833 0.6667 0.7917 0.7083 0.4583 0.5417 0.7083

Columns 22 through 28

0.6667 0.6250 0.6250 0.7083 0.5000 0.6667 0.8333

Columns 29 through 35

0.8750 0.7500 0.8750 0.6667 0.7083 0.5000 0.6667

Columns 36 through 42

0.6667 0.5833 0.6667 0.7083 0.7083 0.4583 0.7500

Columns 43 through 49

0.5000 0.7083 0.7917 0.6250 0.5000 0.7500 0.7917

Columns 50 through 56

0.5000 0.6250 0.6250 0.7083 0.5000 0.4583 0.5833

Columns 57 through 63

0.6250 0.5833 0.6667 0.6667 0.7083 0.6250 0.4167

Columns 64 through 66

0.6250 0.4583 0.6250

El contador pasos por cero en tanto por uno de la cuarta persona arroja los siguientes resultados:

contadorpasosporcero =

Columns 1 through 7

0.3704 0.7778 0.4074 0.3704 0.4444 0.4444 0.9259

Columns 8 through 14

0.6296 0.7037 0.5926 0.5926 0.6667 0.6667 0.5926

Columns 15 through 21

0.6667 0.6296 0.7778 0.8148 0.6667 0.7407 0.7407

Columns 22 through 28

0.5185 0.6296 0.8519 0.7407 0.9630 0.7778 0.5185

Columns 29 through 35

0.5556 0.9259 0.8519 0.7778 0.8519 0.9259 1.0000

Columns 36 through 42

0.8889 0.7778 0.7037 0.5926 0.6667 0.8519 0.5556

Columns 43 through 49

0.8148 0.5556 0.5185 0.6667 0.7778 0.7037 0.8519

Columns 50 through 56

0.7778 0.6296 0.5185 0.4815 0.8148 0.7778 0.5185

Columns 57 through 63

0.9630 0.8889 0.7037 0.5556 0.8889 0.6296 0.8148

Columns 64 through 70

0.6667 0.5556 0.4444 0.4444 0.5926 0.4074 0.6296

Column 71

0.8148

El contador pasos por cero en tanto por uno de la quinta persona arroja los siguientes resultados:

contadorpasosporcero =

Columns 1 through 7

0.3636 0.6364 0.8636 0.6818 0.5909 0.1818 0.1818

Columns 8 through 14

0.1818 0.4091 0.3182 0.5000 0.6364 0.7727 0.5455

Columns 15 through 21

0.5909 0.6364 0.4091 0.5455 0.5909 0.7273 0.6364

Columns 22 through 28

0.5455 0.5455 0.5000 0.3636 0.3636 0.5909 0.6364

Columns 29 through 35

0.4545 0.6818 0.8636 0.9091 0.7273 0.6364 0.7273

Columns 36 through 42

0.6818 1.0000 0.8182 0.7727 0.7273 0.7727 0.6364

Columns 43 through 49

0.6818 0.6364 0.7273 0.6364 0.6364 0.7727 0.7273

Columns 50 through 56

0.5909 0.7273 0.7727 0.7273 0.6818 0.6818 0.6364

Columns 57 through 61

0.7727 0.5909 0.4545 0.6818 0.5455

El contador pasos por cero en tanto por uno de la sexta persona arroja los siguientes resultados:

contadordpasosporcero =

Columns 1 through 7

0.5455 0.8182 0.4091 0.5909 0.6364 0.5909 0.5455

Columns 8 through 14

0.8636 0.6818 0.5455 0.5909 0.5455 0.4545 0.5909

Columns 15 through 21

0.5909 0.4545 0.5000 0.6818 0.5000 0.6364 0.5455

Columns 22 through 28

0.5909 0.6818 0.4545 0.6364 0.5909 0.6364 0.7273

Columns 29 through 35

0.6364 0.9091 0.4091 0.5455 0.7273 0.7273 0.6364

Columns 36 through 42

0.9091 0.8182 0.8182 0.7727 0.7273 0.7727 0.9545

Columns 43 through 49

0.6818 1.0000 0.8636 0.9091 0.7273 0.5909 0.7727

Columns 50 through 56

0.6818 0.6818 0.7727 0.7273 0.7273 0.7273 0.8182

Columns 57 through 63

0.9091 0.9091 0.7727 0.7273 0.5909 0.6364 0.6364

Columns 64 through 70

0.5455 0.8182 0.5909 0.6818 0.7727 0.7273 0.5909

Column 71

0.4545

CONTADOR PASOS POR CERO POR COLUMNA RELATIVO AL MÁXIMO VALOR DE LA MISMA

El contador pasos por cero relativo de la primera persona arroja los siguientes resultados:

contadorpasosporcero'

ans =

Columns 1 through 7

0.3704 0.4815 0.7778 0.9259 0.7037 0.7778 0.7037

Columns 8 through 14

0.8519 0.7778 0.6296 0.7037 0.8519 0.9630 0.9259

Columns 15 through 21

0.8519 0.7037 0.8519 0.8519 0.8889 0.7037 0.7037

Columns 22 through 28

0.7407 0.8889 0.9630 0.8148 0.8889 0.8889 0.6296

Columns 29 through 35

0.4444 0.7407 0.7407 0.6667 0.7407 0.7778 0.8519

Columns 36 through 42

0.8889 0.8148 0.9259 0.7407 0.6296 0.5926 0.7037

Columns 43 through 49

0.8889 0.8519 0.8889 0.7037 0.6667 0.8148 0.8889

Columns 50 through 56

0.9259 0.6667 0.8889 0.8148 1.0000 0.9630 0.6296

Columns 57 through 61

0.5926 0.6667 0.7778 0.7778 0.7407

El contador pasos por cero relativo de la segunda persona arroja los siguientes resultados:

contadorpasosporcero'

ans =

Columns 1 through 7

0.2593 0.2963 0.7037 0.9259 0.5556 0.7778 0.7407

Columns 8 through 14

0.6667 0.5926 0.5926 0.6667 0.7778 0.4815 0.7037

Columns 15 through 21

0.7778 0.5185 0.4815 0.7407 0.7778 0.7407 0.5926

Columns 22 through 28

0.6296 0.5556 0.5926 0.6296 0.6667 0.8148 0.7778

Columns 29 through 35

0.6667 0.8889 1.0000 0.7778 0.9630 0.7037 0.7407

Columns 36 through 42

0.8519 0.8889 0.9259 0.7407 0.5926 0.4815 0.6667

Columns 43 through 49

0.7407 1.0000 0.8889 0.7778 0.9259 0.8148 0.8889

Columns 50 through 56

0.7037 0.7037 0.7407 0.7037 0.7407 0.7778 0.4074

Columns 57 through 61

0.3333 0.4074 0.4815 0.5185 0.4444

El contador pasos por cero relativo de la tercera persona arroja los siguientes resultados:

contadorpasosporcero'

ans =

Columns 1 through 7

0.1600 0.2400 0.5200 0.8400 0.7600 0.6800 0.7600

Columns 8 through 14

0.5600 0.5200 0.4400 0.3600 0.5200 0.6000 0.5200

Columns 15 through 21

0.4000 0.4800 0.6800 0.6000 0.7200 0.8000 0.6800

Columns 22 through 28

1.0000 0.6400 0.7200 0.7200 0.7200 0.6000 0.6000

Columns 29 through 35

0.4800 0.5600 0.6800 0.8000 0.6800 0.8000 0.9600

Columns 36 through 42

0.7600 0.7600 0.8000 0.8000 0.9200 0.8400 0.7600

Columns 43 through 49

0.8400 0.8400 0.7600 0.6800 0.7600 0.8400 0.9600

Columns 50 through 56

0.8800 0.7600 0.7200 0.6800 0.6000 0.5200 0.6800

El contador pasos por cero relativo de la cuarta persona arroja los siguientes resultados:

contadordpasosporcero'

ans =

Columns 1 through 7

0.2500 0.1429 0.5357 0.6071 0.8571 0.8571 0.7143

Columns 8 through 14

0.6071 0.7857 0.8929 0.9643 0.8571 0.6786 0.6071

Columns 15 through 21

0.6429 0.4286 0.6429 0.5714 0.7500 0.7143 0.8571

Columns 22 through 28

0.9643 0.7500 1.0000 0.8571 0.8929 0.7143 0.7857

Columns 29 through 35

0.7857 0.6786 0.7500 0.8571 0.7857 0.7857 0.9643

Columns 36 through 42

0.7857 0.5000 0.5714 0.6429 0.5000 0.7857 0.6071

Columns 43 through 49

0.5000 0.5714 0.5000 0.5714 0.8929 0.3929 0.4286

Columns 50 through 56

0.4643 0.7143 0.3929 0.6429 0.7143 0.3214 0.2500

Columns 57 through 61

0.3571 0.4286 0.4643 0.3571 0.7143

El contador pasos por cero relativo de la quinta persona arroja los siguientes resultados:

contadorpasosporcero'

ans =

Columns 1 through 7

0.2308 0.1538 0.7308 0.8077 0.7308 0.5769 0.6154

Columns 8 through 14

0.5000 0.4231 0.6538 0.4615 0.5385 0.4615 0.5000

Columns 15 through 21

0.4615 0.7692 0.5000 1.0000 0.9231 0.6923 0.5769

Columns 22 through 28

0.7692 0.5385 0.7692 0.7692 0.6154 0.9231 0.6154

Columns 29 through 35

0.6538 0.7692 0.7308 0.6923 0.7308 0.6538 0.6154

Columns 36 through 42

0.7308 0.6923 0.7692 0.8462 0.7308 0.4231 0.5000

Columns 43 through 46

0.2692 0.3462 0.6154 0.5000

El contador pasos por cero relativo de la sexta persona arroja los siguientes resultados:

contadorpasosporcero'

ans =

Columns 1 through 7

0.2857 0.4286 0.7857 0.7143 0.6429 0.7143 0.8214

Columns 8 through 14

0.7143 0.8214 0.7857 0.6071 0.5000 0.8571 0.6071

Columns 15 through 21

0.6786 0.8571 0.6429 0.7500 0.7500 0.8929 0.8214

Columns 22 through 28

0.8929 0.8571 0.6786 0.6071 0.8571 0.7143 0.7857

Columns 29 through 35

0.8929 0.6786 0.6786 0.6786 0.8214 0.8571 0.9286

Columns 36 through 42

0.9643 0.7143 0.9643 0.7857 1.0000 0.8571 0.7500

Columns 43 through 49

0.7857 0.6786 0.6786 0.5000 0.5357 0.4643 0.4286

Columns 50 through 51

0.5357 0.5357

BIBLIOGRAFÍA

1. D. Marr and E. Hildreth, " Theory of edge detection". Proc. R. Soc. London , Ser. B ,207:187-217,1980.
2. G.S. Robinson, "Edge Detection by Compass Gradient Mask," 1977.
3. E.R. Dougherty, An introduction to Morphological Image Processing, 1992.
4. C.R. Giardina and E.R. Dougherty, Morphological Methods in Image and Signal Processing, Prentice-Hall, 1988.
5. W.K. Pratt, Digital Image Processing (2ª edición), John Wiley & Sons, New York, 1991.
6. J.M.S. Prewitt. "Object Enhancement and Extraction"
7. L.S. Davis. " A Survey Of Edge Detection Techniques." Computer Graphics and Image Processing, vol.4, pp.248-270, 1975.
8. W. Frei and C.C. Chen. " Fast Boundary Detection: A generalization of a new algorithm." IEEE Trans Computers 26, nº 2 (October 1977) :988-998
9. P.V.C. Hough. " Method and Means of Recognizing Complex Patterns." 1962
10. Sistemas de percepción y reconocimiento de formas. Apuntes de la escuela de ingenieros.
11. Rafael C. González, Richard E. Woods, "Digital Image Processing", filtrado de Prewitt.

12. JAE S. LIM "Two- Dimensional Signal and Image Processing", Prentice Hall, pag 476-494.
13. Kenneth R. Castleman "Digital Image Processing", filtro Prewitt.
14. Anil K. Jain, "Fundamentals of Digital Image Processing", pp 233-244 y pp 244-255, Prentice Hall.