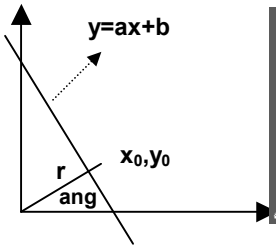
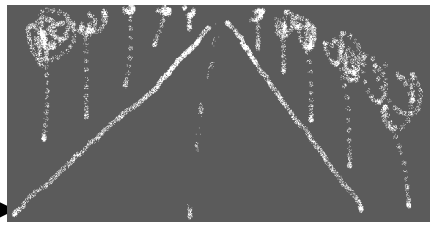
PRUEBA Nº3

Nombre:..... Firma:..... Fecha: 13-08-2007

1.- (10 Ptos.) En una aplicación destinada a controlar la navegación de un robot por una calle, se logró obtener la siguiente imagen de bordes, donde se observan claramente los límites de la calle. Programe la transformada HOUGH para detectar las 2 líneas principales y sus parámetros polares ("r" y "ang").





Según se observa en la imagen ambas rectas deberían ser las mayores concentraciones en el plano de la transformada HOUGH. Proponga una estrategia para encontrar estos dos máximos y así obtener las ecuaciones de ambas rectas.

Ayuda:

Ec. recta polar:

r= x·cos(ang) + y·sin(ang)

Punto de la recta con distancia mínima al origen:

x_0=-ab/(1+a^2), y_0=b-a^2b/(1+a^2)

La transformada HOUGH consisten en calcular la recta polar que une a todos los pares de puntos de la imagen y contabilizar los parámetros de estas rectas en una matriz ("ángulo" y "radio").

```
IM=imread('calle.bmp');
subplot(3,1,1);imshow(IM,[]);
[nf nc]=size(IM)
H=zeros(180+1, round(sqrt(nf^2+nc^2))+1);
for f=1:2:nf
 for c=1:2:nc
  if IM(f,c)==1
   for ff=1:2:nf
    for cc=1:2:nc
     if ff~=f & cc~=c & IM(ff,cc)==1
       %ec recta
       if cc==c, x0=c;y0=0;
       else
         a=(ff-f)/(cc-c);
         b=ff-a*cc;
         x0=-a*b/(1+a*a);
         y0=b-a*a*b/(1+a*a);
       end;
      r=sqrt(x0^2+y0^2)+1;
      if x0==0, ang=90;else ang=atan(y0/x0)*180/pi+90;end;
      H(round(ang),round(r))= H(round(ang),round(r))+1;
     end;%if
    end;%for cc
   end;%for ff
  end;%if
 end;%for c
```

```
end;%foc f

%Se buscan los dos máximos principales
[nang nr]=size(H);
max1=0;
for ang=1:nang
   for r=1:nr
      if H(ang,r)>max1
         max1=H(ang,r);
         ang1=ang;
         r1=r;
      end;
   end;
end;
%Resultado 1, y elimino la zona donde está el máximo
[r1-1 ang1-90]
subplot(3,1,2);image(H);
H(ang1-5:ang1+5,r1-r1*.1:r1+r1*.1)= H(ang1-5:ang1+5,r1-
r1*.1:r1+r1*.1)*0;

max2=0;
for ang=1:nang
   for r=1:nr
      if H(ang,r)>max2
         max2=H(ang,r);
         ang2=ang;
         r2=r;
      end;
   end;
end;
%Resultado 2
[r2-1 ang2-90]
H(ang2-5:ang2+5,r2-r2*.1:r2+r2*.1)= H(ang2-5:ang2+5,r2-
r2*.1:r2+r2*.1)*0
subplot(3,1,3);image(H);
```

2.- (10 Ptos.) Programe un algoritmo (Matlab) en base a la técnica de firmas (signaturas) para analizar y determinar si el objeto en la imagen es un círculo. Suponga que cada imagen binaria posee sólo una figura. Debe determinar la posición de la imagen automáticamente, obtener "la firma" y verificar si corresponde a la firma de un círculo o no. Proponga un criterio para esta verificación. Imagen de ejemplo:
Ayuda Coord. Polares **x = r·cos(ang) y = r·sin(ang)**

```
A=double(imread('circ.bmp'));
subplot(2,2,1);imshow(A,[]);
%BUSCA EL CENTRO
[nf nc]=size(A);
maxf=0; maxc=0;
minf=nf; minc=nc;

for f=1:nf
   for c=1:nc
      if A(f,c)==1,
         if f>maxf, maxf=f;end;
         if c>maxc, maxc=c;end;
         if f<minf, minf=f;end;
         if c<minc, minc=c;end;
      end;
   end;
end;
f0=round((maxf+minf)/2);
c0=round((maxc+minc)/2);
A2=A;A2(f0,c0)=2;
subplot(2,2,2);imshow(A2,[]);

%OBTIENE EL BORDE DOBLE (evita cruces diagonales)
B=double(erode(A));
C=double(dilate(A));
A=(C-A)+(A-B);
subplot(2,2,3);imshow(A,[]);

%OBTIENE FIRMA
ancho=maxc-minc;
alto=maxf-minf;
if alto>ancho, w=alto; else w=ancho;end;
firm=zeros(360,1);
C=A;
for ang=0:359
   r=0;
   while r<w
      x=round(r*cos(ang*pi/180));
      y=round(r*sin(ang*pi/180));
      C(f0+y,c0+x)=1;
      if A(f0+y,c0+x)==1, break;end;
      r=r+1;
   end;
   firm(ang+1)=r;
   subplot(2,2,4);imshow(C,[]);pause(0);
end;

%Si la firma es una línea recta la varianza debería ser muy baja
v=var(firm);
umbral=mean(firm)*0.1;
if v<umbral, disp('Es un círculo'); else disp('No es un círculo');end;
subplot(2,2,4);plot(0:359,firm);axis([0 359 0 w]);
```

3.- (10 Ptos.) Comente: El “crossover” es una técnica cuya efectividad depende de la parametrización del sistema. Indique un ejemplo donde es efectiva y otro donde no lo es.

Los algoritmos genéticos posee 3 "herramientas" fundamentales. La selección de los mejores resultados, la mutación de "genes" y el crossover o entrecruzamiento de los "cromosomas".

Si la parametrización de un problema codifica a cada cromosoma con genes irrepetibles, entonces el entrecruzamiento se debería realizar al interior de un mismo cromosoma y NO entre cromosomas.

Por ejemplo para el problema del vendedor viajero, si se codifica a cada cromosoma como la secuencia de ciudades que se deben recorrer, no tiene sentido entrecruzar un cromosoma C1=1,2,3,4,5,6,7,8 con otro C2=8,7,6,5,1,2,3,4, puede dar un resultado incoherente para la función de "premio". Por ejemplo si se entrecruza en a partir del quinto gen el resultado sería: C3=1,2,3,4,1,2,3,4 lo cual no tiene sentido. La solución es generar entrecruces al interior de un mismo cromosoma.

Un ejemplo donde sí tiene sentido el entrecruzamiento es cuando se utilizan algoritmos genéticos para seleccionar las mejores redes neuronales para un problema dado. Si se entrecruzan dos redes neuronales existe la posibilidad que al intercambiar parte de sus pesos se genere una nueva red neuronal cuyo resultado sea mejor. En este cado la parametrización del problema no se afecta si se intercambian pesos de una red a otra, es decir de un cromosoma a otro.

4.- (10 Ptos.) En compresión de imágenes, ¿cuál es la diferencia entre “eliminar código redundante” y “eliminar píxeles redundantes”?

Eliminar código redundante se refiere a tratar de disminuir la cantidad de bits asignados a cada símbolo. Por ejemplo una codificación de 1 byte por píxel para una imagen binaria no es óptima, pues en vez de asignar 8 bit por cada píxel es posible asignar 1 bit por píxel, dado que una imagen binaria posee píxeles con sólo 2 valores posibles.

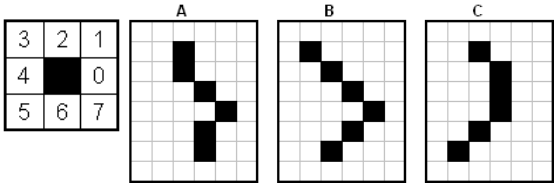
Eliminar píxeles redundantes se refiere a la posibilidad de que existan símbolos o píxeles repetidos al interior de una misma imagen. Si la posición de dichos píxeles se puede codificar, entonces se puede almacenar sólo una vez el valor del símbolo o píxeles y agregar la codificación de la posición. Por ejemplo si una imagen tiene una fila con 256 píxeles blancos, en vez de almacenar 256 byte sería posible almacenar 1 byte con el color y 1 byte con la cantidad de veces que se repite dicho color en la fila.

5.- (10 Ptos.) Se desea utilizar el método basado en “códigos de cadena” para comparar las tres figuras. Determine cuál figura B o C se “parece más” a la figura A, en base a al coeficiente de correlación de las cadenas.
Ayuda: cos(45º)=0.7

$$C_{ab} = \frac{1}{n}\sum_{i=1}^{n} a_i b_i$$

Donde:

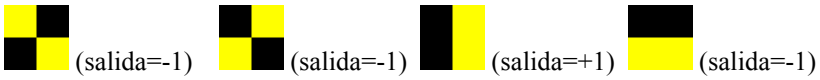
$$a_i b_i = \cos(\angle a_i - \angle b_i)$$



Cadena A
6-7-7-5-6
Cadena B
7-7-7-5-5 = (cos45+cos0+cos0+cos0+cos45) = 0.7+1+1+1+0.7 = 4.4 ==> 4.4/5
Cadena C
7-6-6-5-5 = (cos45+cos45+cos45+cos0+cos45) = 0.7+0.7+0.7+1+0.7 = 3.8 ==> 3.8/5

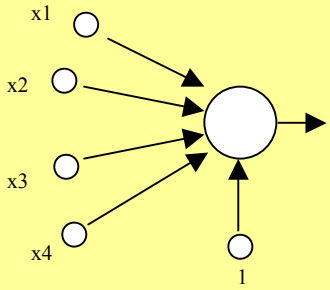
El más parecido a la cadena A es la cadena B.

6.- (10 Ptos.) Programe un perceptrón (1 neurona), para "reconocer" líneas con pendiente 90º (como el patrón número 3). Realice manualmente la primera actualización de los pesos. Al inicio, suponga todos los pesos en 0. Entrene ingresando los patrones en el mismo orden:



Patrones: 0110|-1, 1001|-1, 1010|+1, 1100|-1

Unicialmente todos los pesos están en "0".



w1=0
w2=0
w3=0
w4=0
w0=0

Se inicia el proceso de entrenamiento:

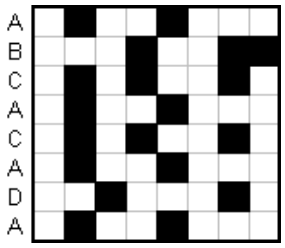
Patrón Nº1
x1=0, x2=1, x3=1, x4=0, Clase= -1
Se ejecuta la red neuronal:
y=F(w1*x1+w2*x2+w3*x3+w4*x4+w0)
y=F(0) = -1 ==> Clasifica Bien!

Patrón Nº2
x1=1, x2=0, x3=0, x4=1, Clase=-1
Se ejecuta la red neuronal:
y=F(w1*x1+w2*x2+w3*x3+w4*x4+w0)
y=F(0) = -1 ==> Clasifica Bien!

Patrón Nº3
x1=1, x2=0, x3=1, x4=0, Clase=+1
Se ejecuta la red neuronal:
y=F(w1*x1+w2*x2+w3*x3+w4*x4+w0)
y=F(0) = -1 ==> Clasifica MAL! ==> Actualización de pesos

w1=w1'+d(patron)*x1 = 0 + (+1)*1 = 1
w2=w2'+d(patron)*x2 = 0 + (+1)*0 = 0
w3=w3'+d(patron)*x3 = 0 + (+1)*1 = 1
w4=w4'+d(patron)*x4 = 0 + (+1)*0 = 0
w0=w0'+d(patron) * 1 = 0 + (+1)*1 = 1

7.- (10 Ptos.) Suponga la siguiente imagen binaria de 8x8 píxeles. Considere cada fila de la imagen como 1 byte. Utilice método Huffman para codificar los 8 bytes que posee la imagen.



A) ¿Cuántos bits se deberán transmitir utilizando compresión Huffman y cual es la razón de compresión que se logra?

B) Calcule la entropía de la imagen y compare este valor con los bits por símbolo promedio resultantes de la codificación Huffman. Comente

$$H = -\sum_{i=1}^{I} P(a_i) \log_2 P(a_i)$$

Ayuda: $\log_2(1/8)=-3$ $\log_2(2/8)=-2$ $\log_2(4/8)=-1$

a)
Existen 4 símbolos de 8 bits.

B(1) D(1) C(2) A(4)

(2) C(2) A(4)
B(1) D(1)

(4) A(4)
(2) C(2)
B(1) D(1)

(8)
(4) A(4)
(2) C(2)
B(1) D(1)

Codificación:

A=1
C=01
D=001
B=000

Transmisión Total= (A)1*4+(B)3*1+(C)2*2+(D)3*1=14 bits.

Razón de Compresión: 64:14 = 4.5 veces

Probabilidades:
Pa=4/8;
Pb=1/8;
Pc=2/8;
Pd=1/8;

Entropía= - Pa*$\log_2$(Pa) - Pb*$\log_2$(Pb) - Pc*$\log_2$(Pc) - Pd*$\log_2$(Pd) = +4/8 + 3/8 + 4/8 + 3/8 = 14/8 = 1.75

Promedio de bits por símbolo en codificación Huffman = (1+2+3+3)/4 = 9/4 = 2.25

El valor de la entropía de la imagen 1.75, es menor al logrado por la codificación Huffman. Esto es correcto pues la entropía es el límite teórico máximo de compresión. Huffman se acerca a dicho valor. Sin duda, 2.25 es mejor que la codificación sin compresión con valor de 8 bits por píxel.