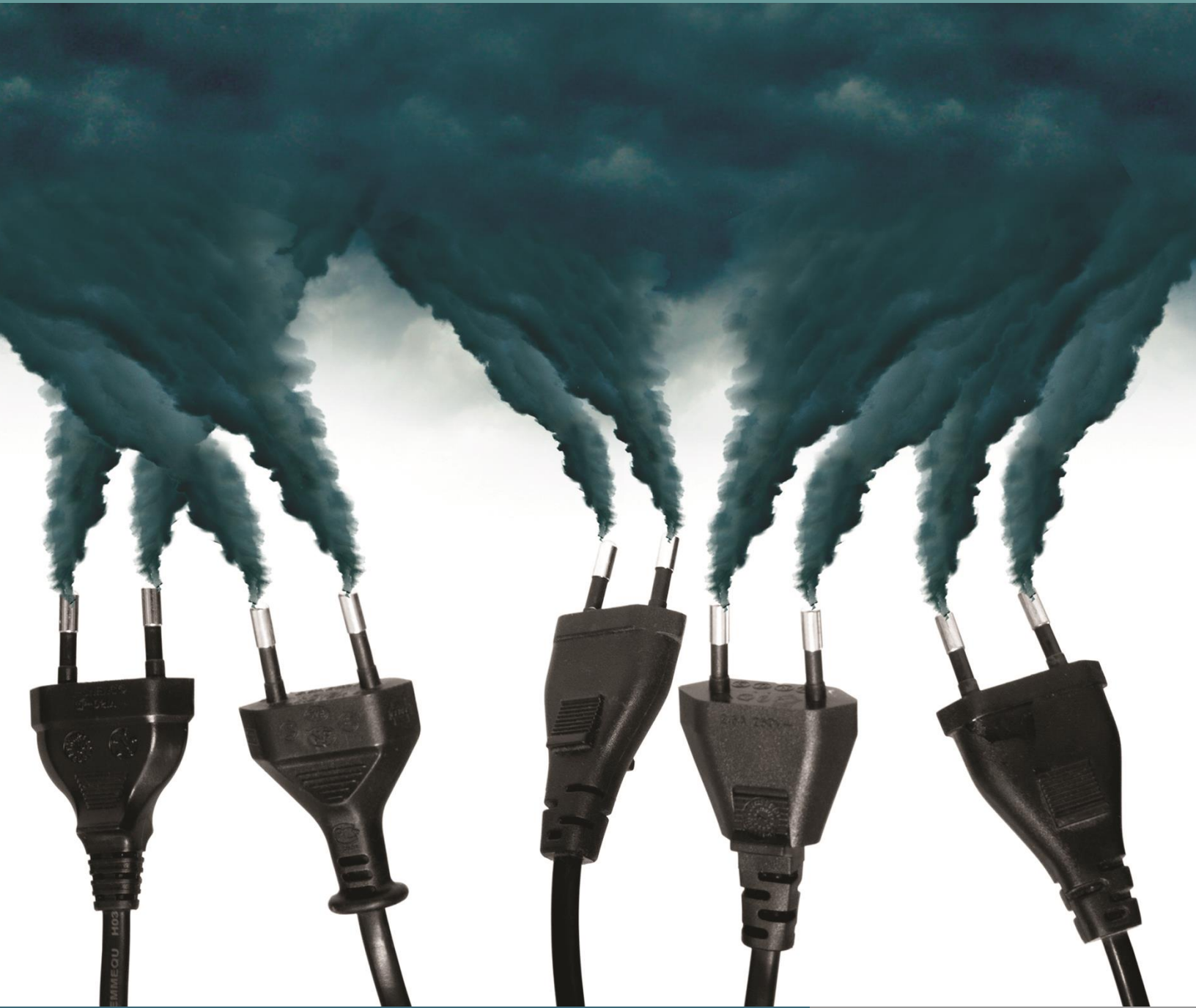


Códigos CCO & FUZZY



Avance Tesis

Luixandra Azereth Saucedo Quirino

UDG
30/05/2022

1. CCO

Para comenzar a entender el funcionamiento del código CCO, primeramente, se analizaron los diferentes códigos que componen el CCO y se investigo acerca de su funcionamiento.

Posteriormente se corrió el código dándole como valores de entrada los siguientes:

Donde N_p = numero de partículas, $MaxGeneration$ = iteraciones, dim = numero de umbral (en este caso desde 2 umbrales en adelante)

$[x_{best}, f_{obj}] = CCO(3, 1000, 2)$ = los valores requeridos por la siguiente function;

```
function [fobj,xbest,imgOut] = CCO(Np,MaxGeneration,dim)
```

el código nos pide los siguientes valores para su inicio

command window:

```
Np=50;
```

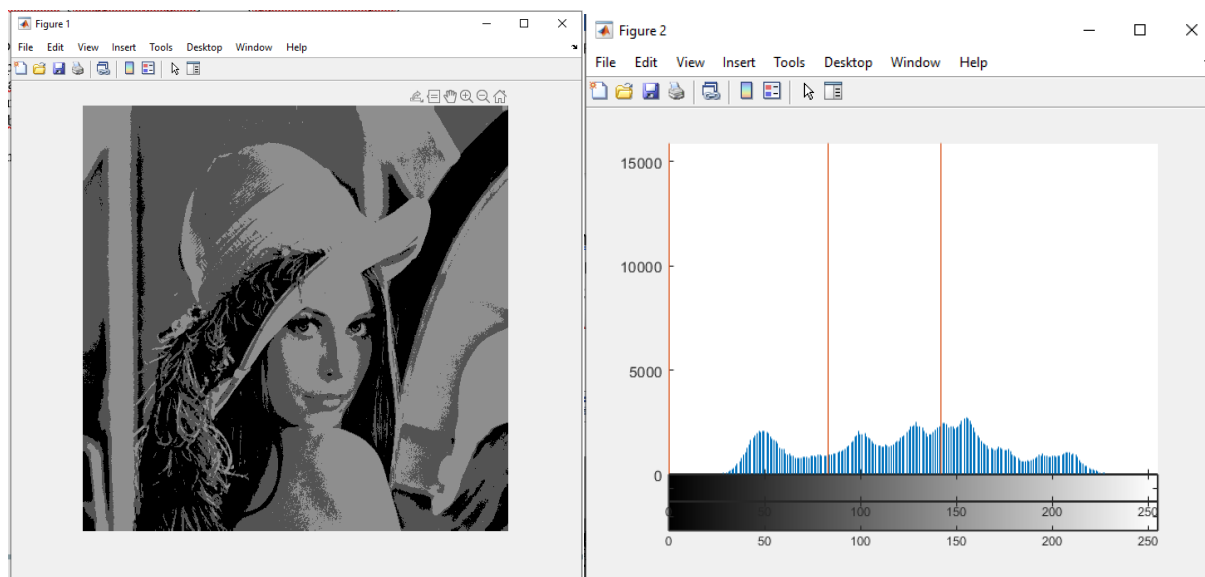
```
MaxGeneration=50000;
```

```
dim=2;
```

En la primera prueba se modificaron por los siguientes:

```
[xbest,fobj]=CCO(3,1000,2);
```

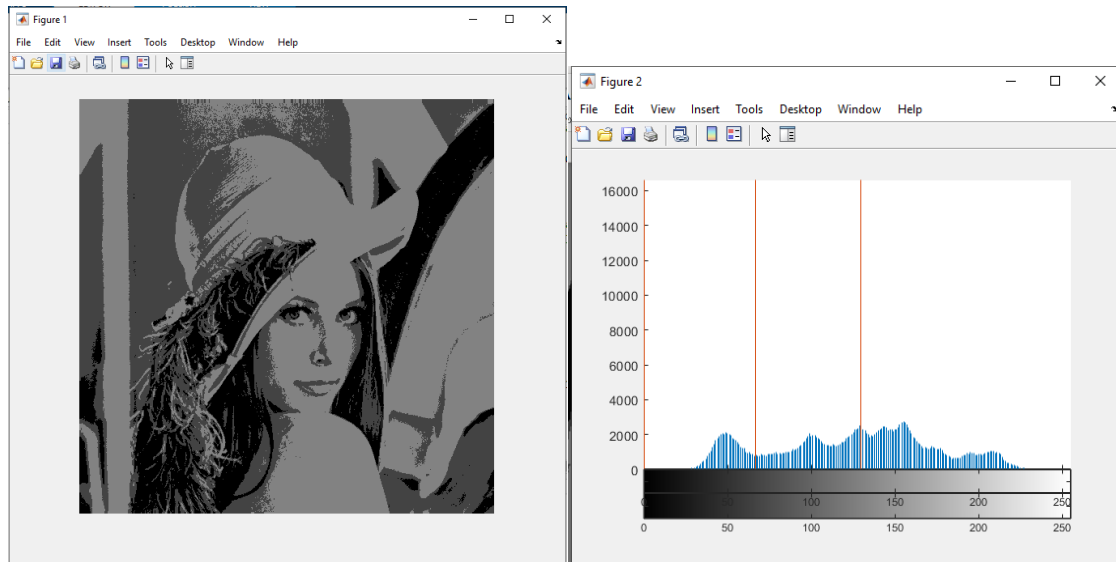
Una vez ejecutado el código nos desplegó lo siguiente:



En donde se puede observar la imagen “lena” ya segmentada en el histograma se observan la escala de grises que conforman la imagen y dos líneas que nos indican los umbrales en los cuales nos segmentara la imagen.

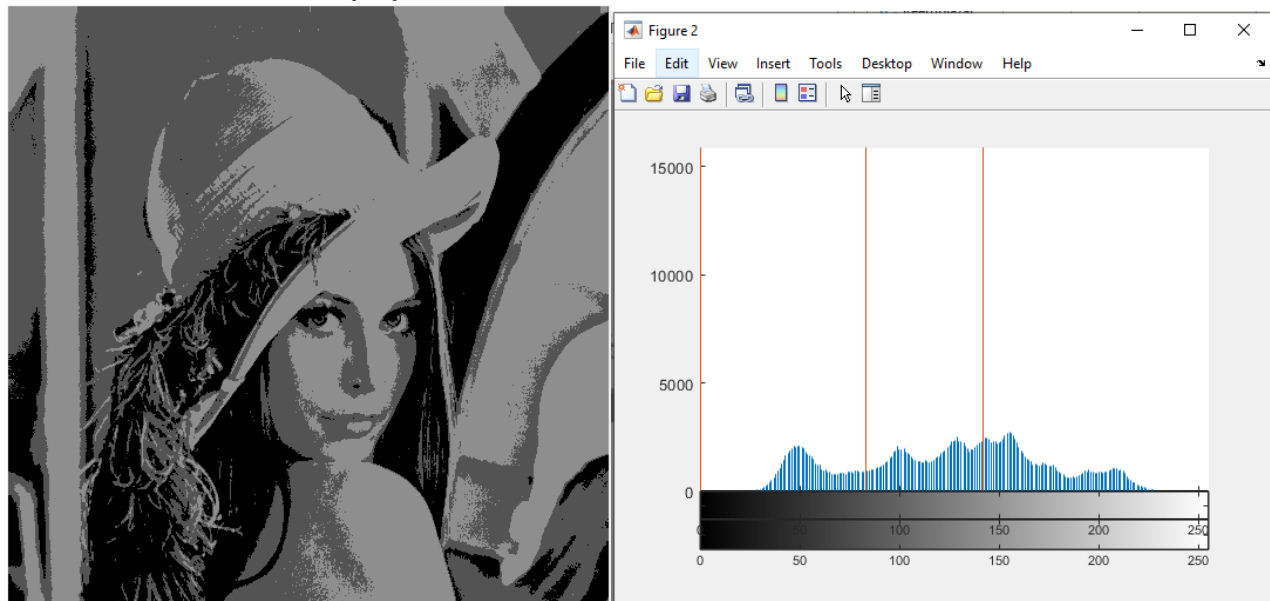
Despues se disminuyo el numero de MaxGeneration de 1000 a solo 10 para observar su comportamiento.

Se obtuvo una imagen mucho menos especifica y una grafica con los umbrales especificados



En caso contrario al subir el numero de Max Generation a 10000 nos entrega una imagen mucho mas definida en cuanto a la escala de grises.

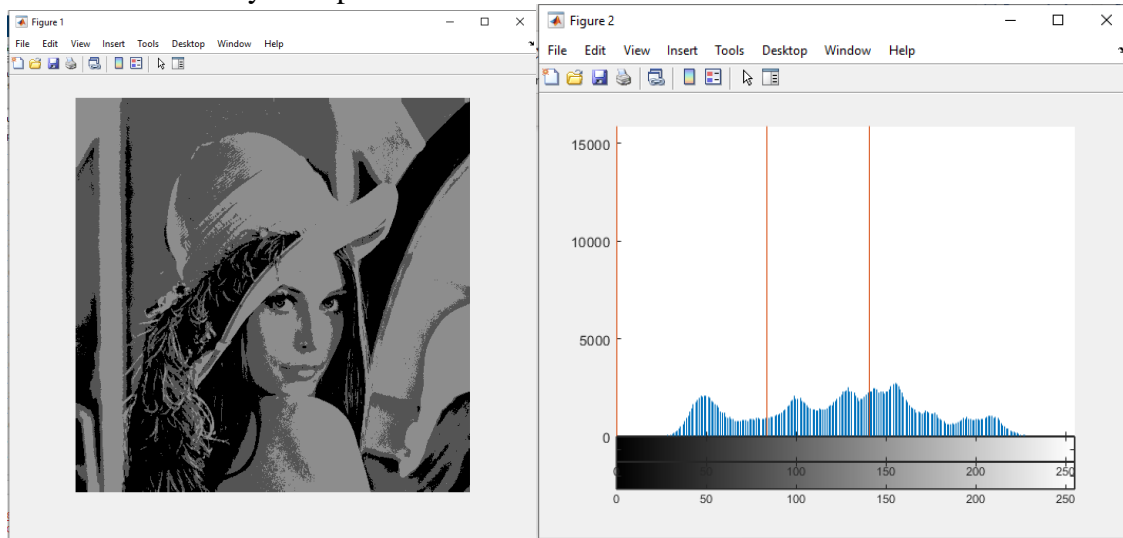
En la cual si observamos detenidamente, se notaran los detalles mas finos, como lo es en la parte de la nariz, el cabello y ojos, entre los mas sobresalientes.



Ahora se modificaran el valor de el numero de particulas “Np”

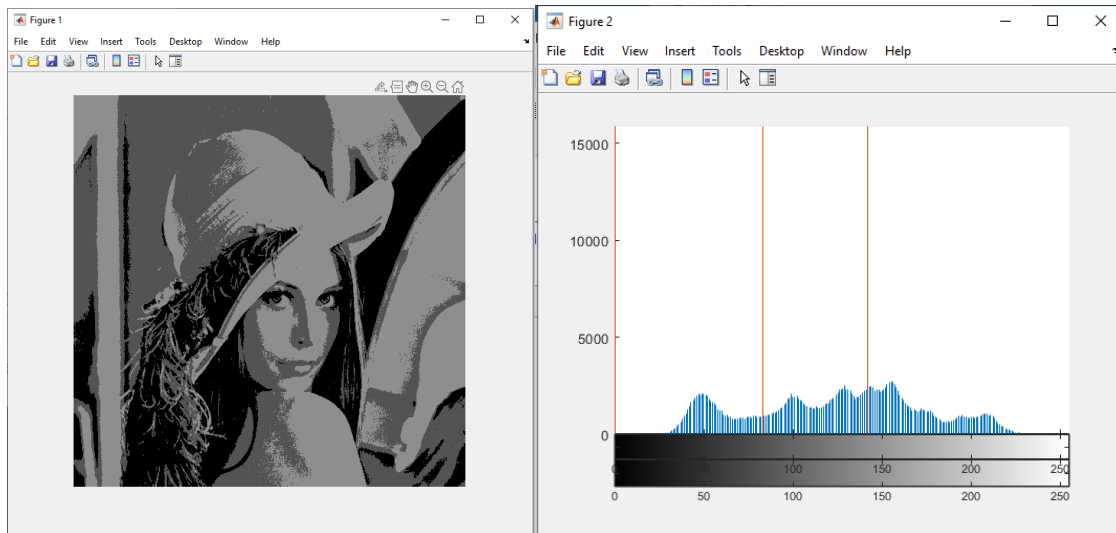
```
[xbest,fobj]=CCO(1000,1000,2);
```

Al tener un numero mayor de particulas



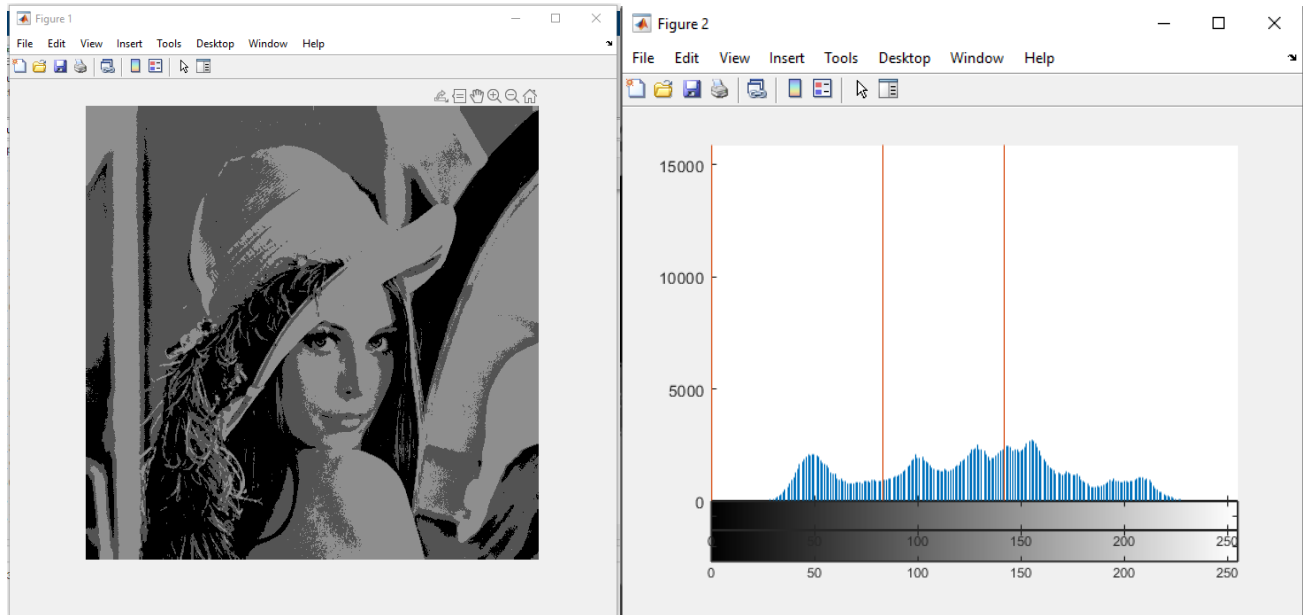
Despues se disminuiran a solo 10 para hacer mas notorio el funcionamiento de dichas particulas:

```
[xbest,fobj]=CCO(1000,1000,2);
```



En esta ocasion se modificaran los umbrales en “dim” comenzando con 2:

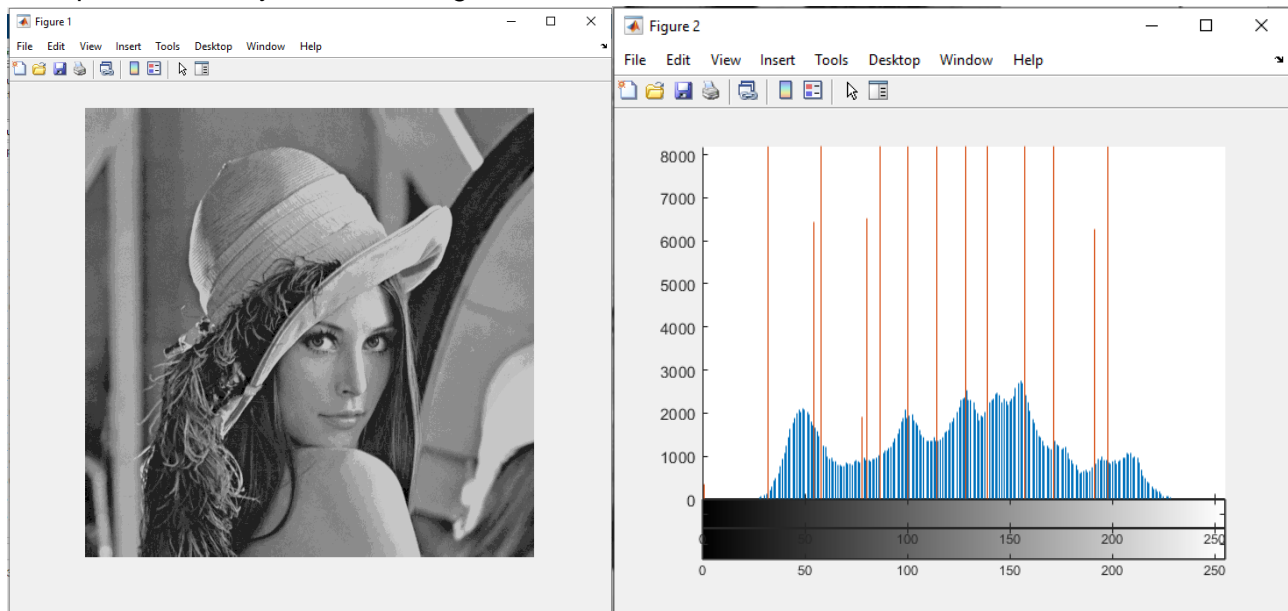
```
>> [xbest,fobj]=CCO(3,1000,2);
```



Despues se comparo con una modificacion por un numero mayor 15

```
>> [xbest,fobj]=CCO(3,1000,15);
```

En este resultado se observa de una manera mucho mas visible las diferencias en la imagen, en la imagen anterior donde solo se tenian 2 segmentaciones la imagen estaba mucho mas borrosa y menos definida, en este caso al aumentarse en 15 podemos observar en el histograma en que puntos se realizaron las segmentaciones, al ser un numero mas amplio cada area se distingue con mayor facilidad que la anterior, obteniendo una imagen con una variedad de grises mucho mayor, hacienda visiblemente posible apreciar las mejoras en la imagen.





































































2. FUZZY

En este código la dinámica fue diferente al CCO, ya que el primer código solo era cuestión de analizar su funcionamiento y dar parámetros de inicialización, en cambio en fuzzy primeramente se analizó cada uno de los códigos y funciones que lo componen para después analizar como se llamaba a cada una de las funciones y archivos de este.

Una vez comprendidas las etapas que manejaba el código y sus funciones se creó una carpeta nueva donde se modificaron los códigos y archivos para obtener un nuevo código de Fuzzy que a diferencia del otro funcione con MCE como lo hace el método del CCO.

Primero que nada en la carpeta de fuzzy se importaron los archivos que se necesitarían del método CCO, los cuales fueron los siguientes

CCO	FUZZY	MODIFICADO
 CCO  centroide  chaos  chaosVector  clusterDensity  evaluateF  evolutionProcess  findrange  histogramaSegmentacion  imageGRAY  imageGRAY2  ImagenSegmentada  initialization  iterador  iteradorfo  lena  minim  Multi_MCE  PSNR  test_functions  vTransform	 50dimTestData  100dim5corridasFuzzy  comp  distMax  Eudistancia  evaluateF  findrange  function_range  fuzzyL  fuzzyL2  initialization  mfDistancia  mfRango  simplebounds  t0  test  test_functions  test_functions_range  tFuzzy  tsugeno	 50dimTestData  100dim5corridasFuzzy  comp  distMax  Eudistancia  evaluateF  findrange  function_range  fuzzyL.asv  fuzzyL  fuzzyL2  imageGRAY2  initialization  lena  mfDistancia  mfRango  Multi_MCE  Reporte para Git. Dr.Wario, Dr. Jorge  simplebounds  t0  test  test_functions  test_functions_range  tFuzzy  tsugeno

Una vez que se organize la nueva carpeta se realizaron algunas modificaciones en el código para que fuera capaz de funcionar con MCE.

El primer archivo que se modificó fue el “evaluate F” se tomó del CCO y se colocó en la nueva carpeta

Dentro del archivo “FuzzyL” se modificó lo siguiente:

Primamente se colocaron las siguientes líneas para poder leer la imagen:

```
I=imread('lena.tif');  
Iseg=imread('lena.tif');  
hist=imhist(I);  
[n,m] = size(I);  
hist = hist / (n * m);
```

Después se en el mismo archivo se le dio un valor a F_Index debido a que fuzzy los maneja del 1 al 20, se seleccionó el 7.

```
F_index=7;
```

en el archivo “test_functions_range” se modificaron los valores de F_index ==7 los cuales se encontraban en: F_index==7 down=-1.28; up=1.28;

```
if F_index==7  
    down=1; up=256;  
end
```


Antes de hacer esta modificación y si se corria “FuzzyL” se generaban diferentes errores que nos enviaban al código “Multi_MCE.m” donde el primer error nos indicaba que los valores debían ser números enteros y positivos.

Y al ocurrir este error, se detenía la ejecución y no era posible continuar.

Después nos generaba un Segundo error el cual nos decía que m1 y m2 no podían exceder en 1.

Pero estos errores fueron corregidos al realizar el cambio mostrado en la imagen anterior.

```
else
    v1 = T(ii-1);
    v2 = T(ii)-1;
end

for i = v1:v2 % miu from 1 to tn
    m1 = m1 + (i * hn(i));
    m2 = m2 + hn(i);
end

if m2 == 0
    miu(ii)= m1 / (m2+eps);
else
    miu(ii)= m1 / m2;
end
```

Después el archivo “test_functions” se tomó del CCO y se agregó a la carpeta con el archivo modificado

```
% This function calculates the value of the objective function
function fit=test_functions(L,h)

% %Displaced fitness function f01
%up = ones(1,dim) .* up;
%L = up - L;
%fit=sum(L.^2);

fit=Multi_MCE(L,h);
```

Se le agregó al inicio de “FuzzyL” una línea de `close all` para poder limpiar la entre cada nueva ejecución ya que al correrlo nuevamente siempre era necesario poner un `close all`;, `clear all`;, `clc`; y de esta manera se realiza de manera automática.


```
%Fuzzy Modificado%
close all function [xbest,fobj,X,fit] = fuzzyL(F_index)
```

En este mismo archivo “FuzzyL” se modifico la siguiente linea:

```
fitnessAUX=evaluateF(X,F_index);
```

en la cual se cambio el evaluateF por evaluateF(X,hist);

donde hist es el histograma , en esta parte del codigo me generaba error el cual me dio bastante problema hasta notar que el histograma estaba definido con la letra “h” pero en este mismo codigo la “h” ya tenia otros valores asignados por los que al correr el codigo leeia la “h” incorrecta, por lo que se opto por modificar el nombre del histograma y cambiarlo de “h” a “hist”.

```
%modif h
fitnessAUX=evaluateF(X,hist);
[Lightn,Index]=sort(fitnessAUX,'ascend');
xbest = X(Index(1),:);
fobj=Lightn(1);
fmax=Lightn(end);
%Y=fitnessAUX;
```

Despues dentro de la grafica de la población inicial se modico primeramente el nombre de la figura de “figure(1)” a “figure(3)”.

```
figure(3) %Gráfica de la población inicial
hold on
```

Tambien al momento de calcular z, aparecia un error.

La “z” es calculada debido a que es necesaria para poder graficar.

```
z(h,1)=evaluateF([xg(h,1),yg(h,1)],F_index);
```

Una vez modificada se cambio “F_index” por el histograma.

```
for l=1:col
    for h=1:row
        z(h,1)=evaluateF([xg(h,1),yg(h,1)], hist);%
```

En el siguiente `while` $i \leq 50$ se bajo a 50 ya que se encontraba en 5000 y se disminuyo para las pruebas.

Al llegar a “R” nos lanzo un error mas debido a que tenia el `F_index` como paso en “z” y nuevamente se modifico por “hist”

```
%se quita fmax, evaluateF(X(j,:))
R = tsugeno(beta,beta1,Per,dim,xbest,X(j,:),up,low,fobj,fmax,evaluateF(X(j,:), hist));
```

En este mismo archivo en la parte de “fitnessAux_2” se modificaron las siguientes dos lineas:

```
fitnessAUX_2=evaluateF(X2(1,:) ,F_index);
fitnessAUX = evaluateF(X(j,:),F_index);
```

en las cuales como en “z” y “R” se cambio “`F_index`” por “hist”

```
fitnessAUX_2=evaluateF(X2(1,:) ,hist);
i = i+1;
% eliminar evaluateF(X(j,:))
fitnessAUX = evaluateF(X(j,:),hist);
```

“`F_index`” Tambien fue necesario cambiarlo por “hist” al llegar al findrange “X” quedando de la siguiente manera:

```
X = findrange(X,low,up,dim,Np);

fitnessAUX=evaluateF(X,hist);
[Lightn,Index]=sort(fitnessAUX,'ascend');
i = i+50;

if (Lightn(1) < fobj)
    xbest = X(Index(1),:);
    fobj=Lightn(1);
end
```

Al llegar a la parte del código donde se calculó “zz” al igual que en “z” se cambió F_index por “hist”.

```
for q=1:col1
    for hh=1:row1
        zz(hh,q)=evaluateF([X1(hh,q),Y1(hh,q)], hist);
        %zz(hh,q)=mifit([X1(hh,q),Y1(hh,q)],funcion);%Se
    end
```

Se agregó también a la nueva carpeta el archivo del código “ImageGRAY2” y la imagen con la que se realizarían las pruebas en este caso “lena.tif”

Después de volver a correr el código no se obtenía la imagen, y se notó el error al haberla inicializado pero no se tenía salida por lo que se colocó la imagen de salida, así como también la del histograma.

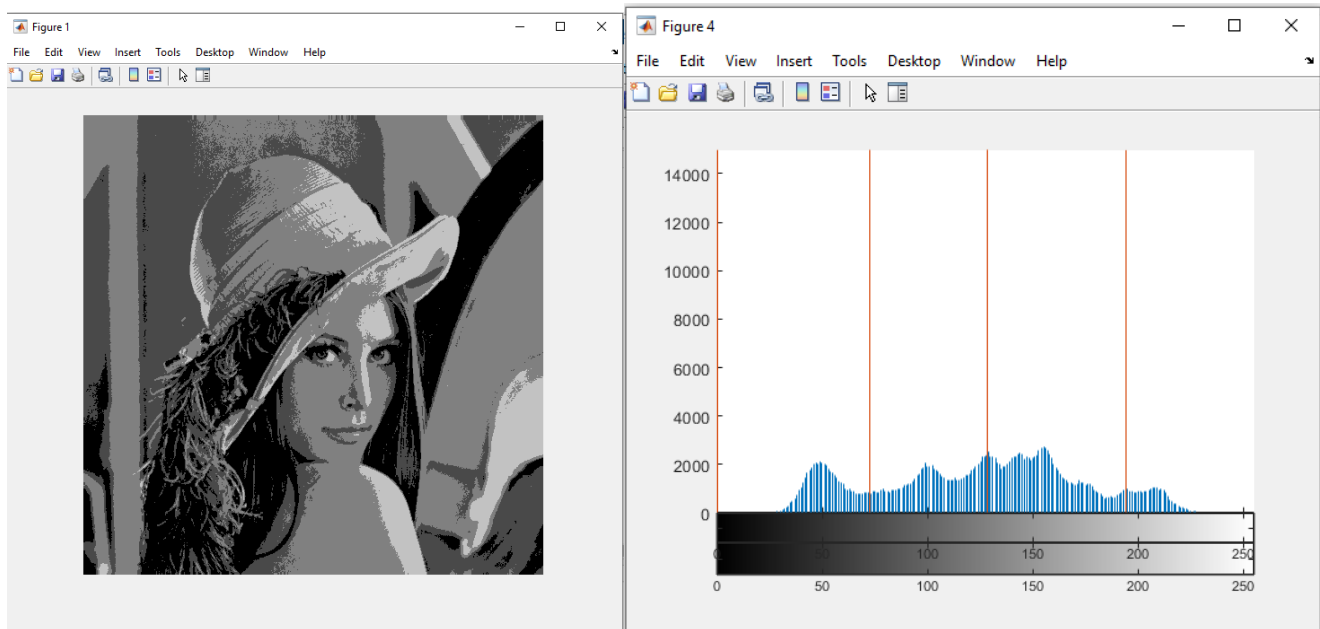
```
xbest=fix(sort(xbest));

imgOut=imageGRAY2(Iseg,xbest); %Coloca los valores 0 y 255 a los extremos
% imgOut=imageGRAY(Iseg,xbest); % No coloca los valores de 0 y 255
imshow(imgOut);
|
figure(4)
hold on;
imhist(Iseg,400);
imhist(imgOut,600);
hold off;
```

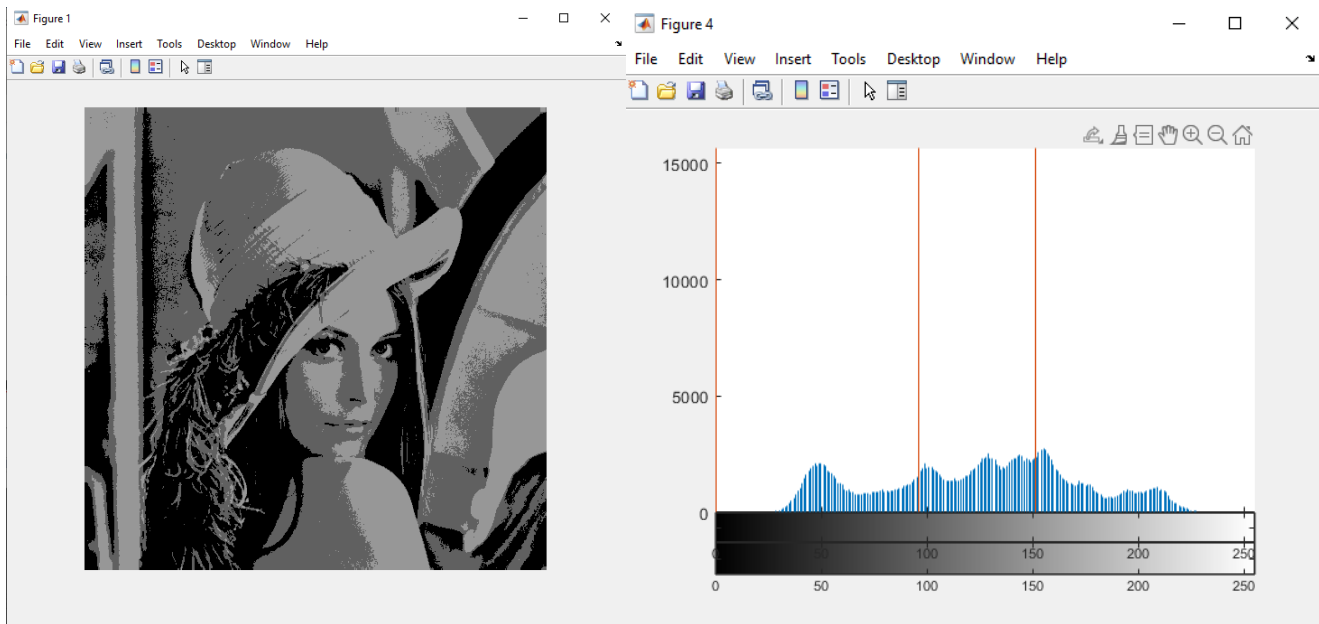
Se volvió a correr nuevamente el código y en esta ocasión ya era posible visualizar la imagen y las gráficas pero los valores de “xbest” no se encontraban redondeados y salían desordenados por lo que se buscó una función que solucionara el problema y se utilizó “fix” y “sort” para redondearlos y ordenarlos con la siguiente línea:

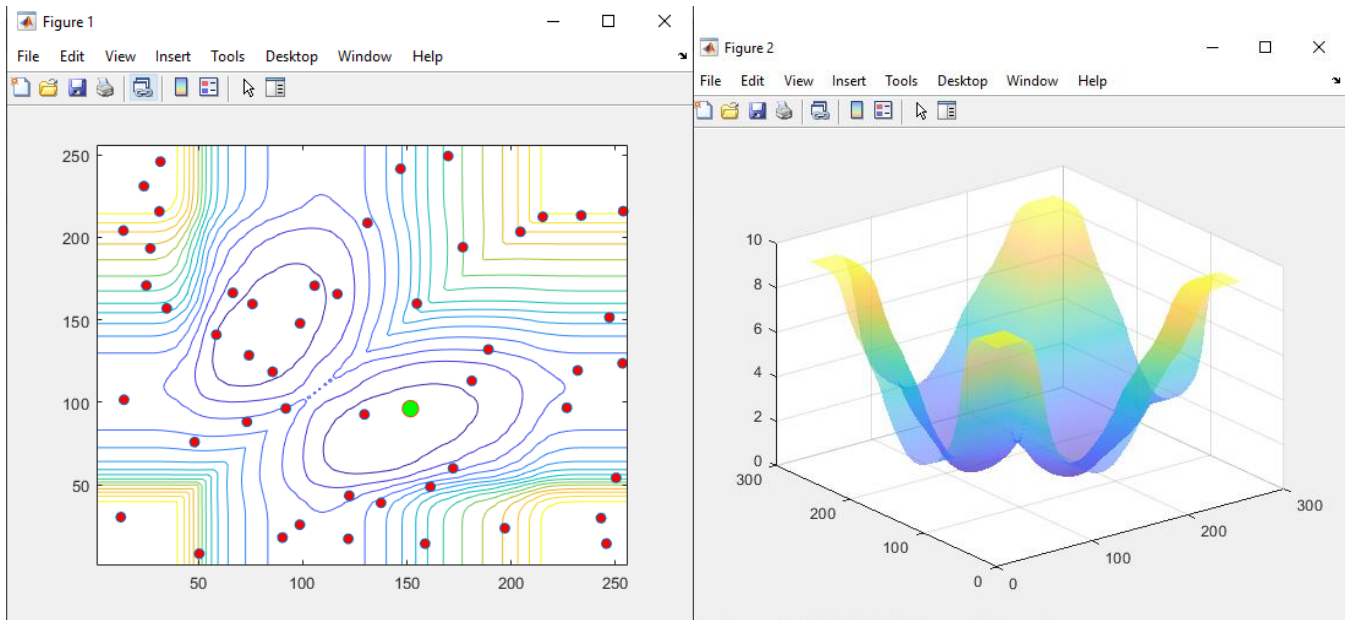
```
xbest=fix(sort(xbest));
```

Se corrio nuevamente obteniendo la siguiente grafica e histograma:



Y si se cambia el “dim” a solo 2 se obtienen lo siguiente:





Al momento de realizar las pruebas al nuevo código se observó que en la parte de “R” Si se sustitúan el número asignado a las R que anteriormente se encontraban en 1,2,3,4 y se colocaban todas en “4” el código funcionaba de una mejor manera

```
R = tsugeno(beta,beta1,Per,dim,xbest,X(j,:),up,low,fobj,fmax,evaluateF(X(j,:), hist));

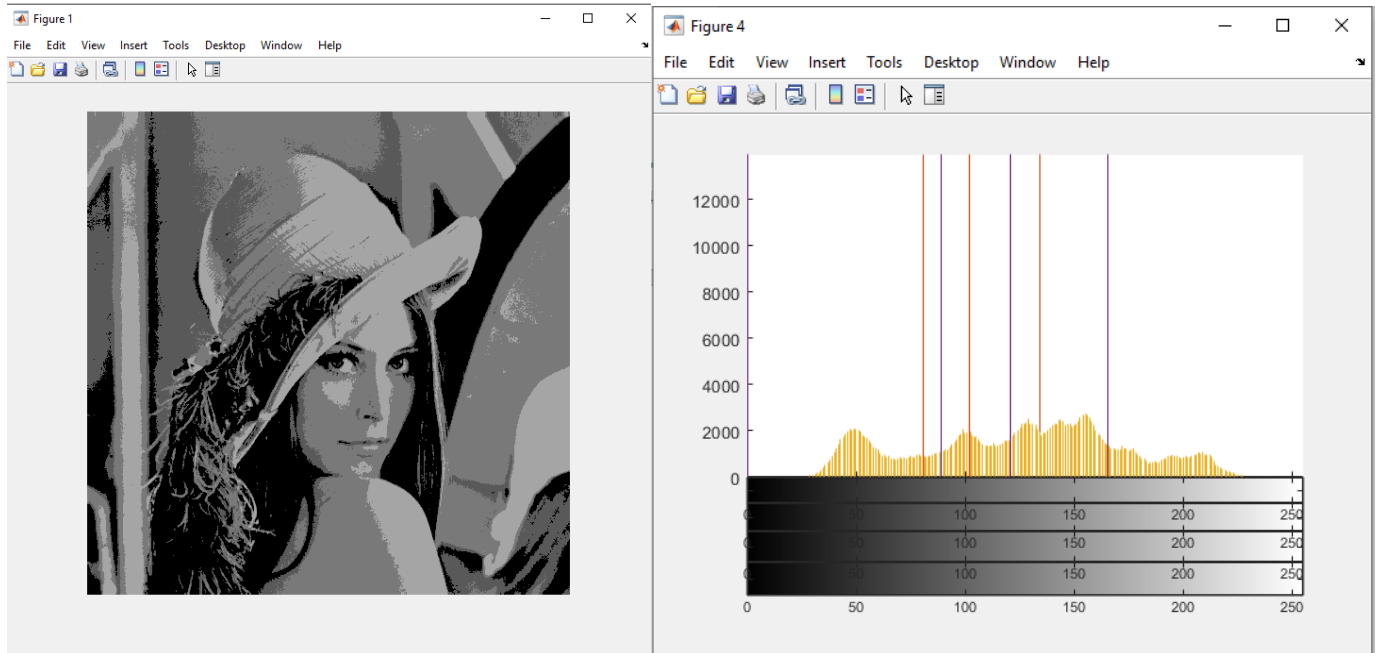
for k = 1:dim
    %Temp(1,k) = ((w1 * R(k,1) + w2 * R(k,2) + w3 * R(k,3) + w4 * R(k,4)) / (w1+w2+w3+w4))*Tun;
    %Temp(1,k) = ((w1 * R(k,4) + w2 * R(k,4) + w3 * R(k,4) + w4 * R(k,4)) / (w1+w2+w3+w4))*Tun;
    %X(j,k) = X(j,k) + (Temp(1,k));
    X2(1,k) = X(j,k) ;
end
```

Para realizar esta modificación también fue necesario entrar al código del archivo “tsugeno”

Donde se modificó el “dim” por “4”

```
R4 = initialization(4,1,up,low);
for i = 1 : dim
```

Al correrlo se obtuvo la siguiente imagen:



Avances Código CCO y Fuzzy