Internet of Things

Cada día más estamos inundados de datos procedentes de muy diferentes sensores. Se capaz de adquirir y comprimir los datos sin perder demasiada información resulta muy importante.

Enunciado:

Se trata en este caso práctico de ser capaz de comprimir unos datos sin que se produzca una pérdida importante de información. La información que se ha recogido han sido audios de vehículos circulando por una calle. El programa, debe probar diferentes opciones de energía para comprimir los datos con una transformada de coseno discreta, mostrando en pantalla la reconstrucción realizada en cada caso (en que nos quedamos con una serie de coeficientes, no todos), así como el error que se produce con cada reconstrucción y el porcentaje de datos (coeficientes) que podemos usar para transmitir o almacenar la información.

Implementación:

`audio1.m`: script principal para seleccionar ficheros ejemplos y ver resultados

`coche1.mp4, coche2.mp4`: ficheros de audio de un coche circulando por un carril, y de dos coches que se cruzan, respectivamente

COMPRESIÓN Y DESCOMPRESIÓN DE AUDIOS DE VEHÍCULOS

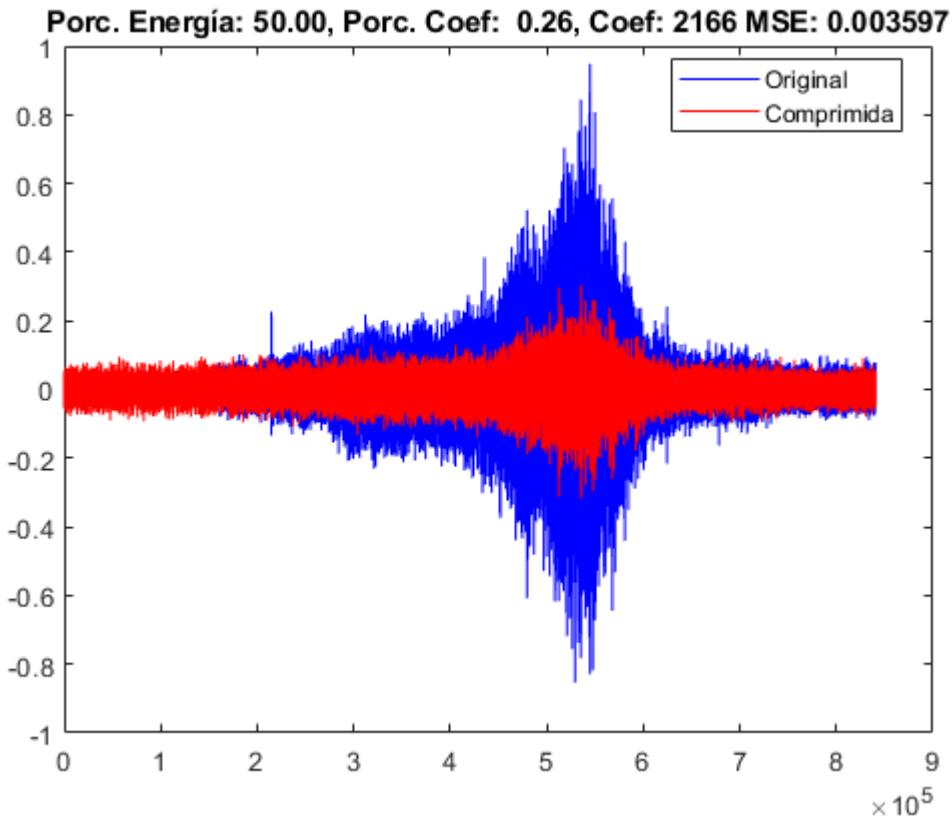
Lectura de datos de audio

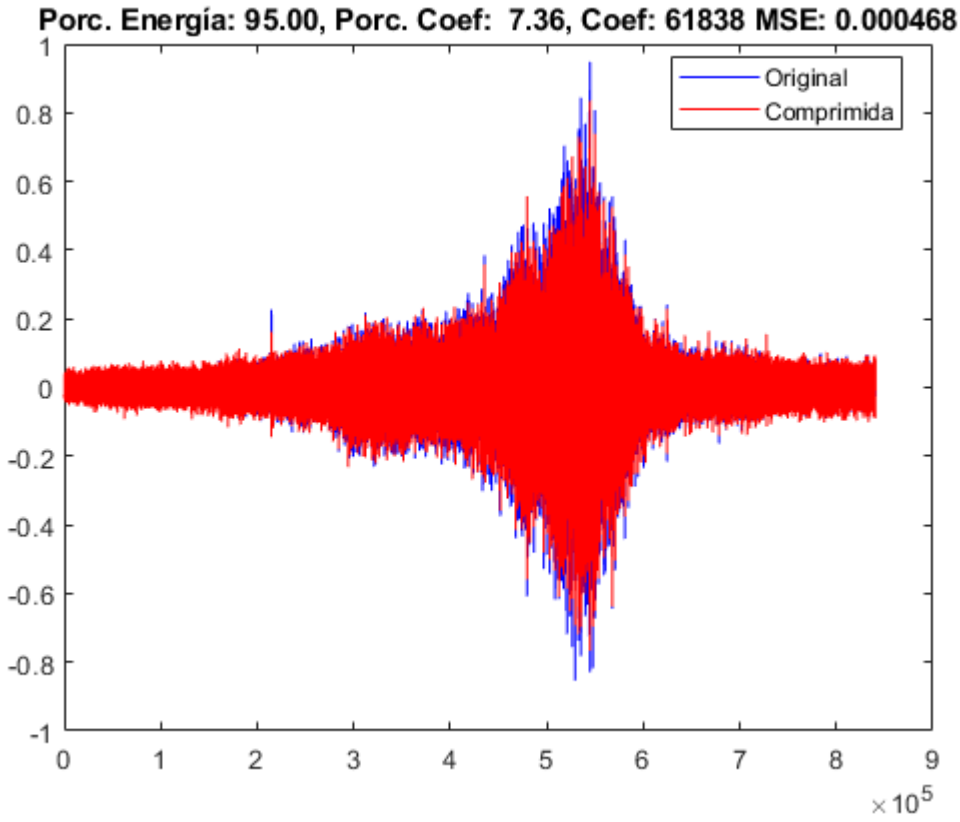
```
clear all;
close all;
clc;

% seleccionamos la señal
disp('Indica qué señal deseas analizar ');
disp('    1.- un coche circulando en un sentido');
disp('    2.- dos coches, en dos sentidos');
n = input('Introduce 1 ó 2: ');

if n == 1
    [y1,Fs1] = audioread('coche1.mp4'); % un coche
    x = y1;
    Fs = Fs1;
else if n == 2
        [y2,Fs2] = audioread('coche2.mp4'); % dos coches (uno en cada sentido)
        x = y2;
        Fs = Fs2;
    else
        error('Intente de nuevo');
    end
end
```

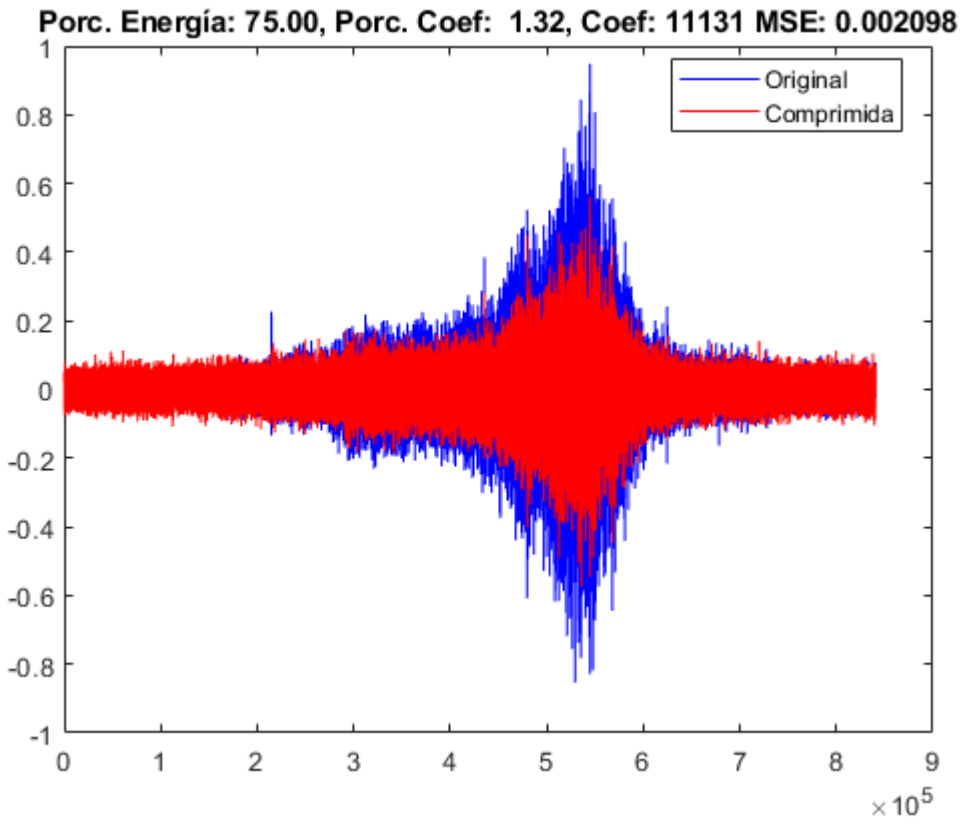
```
Indica qué señal deseas analizar
    1.- un coche circulando en un sentido
    2.- dos coches, en dos sentidos

Introduce 1 ó 2:
```

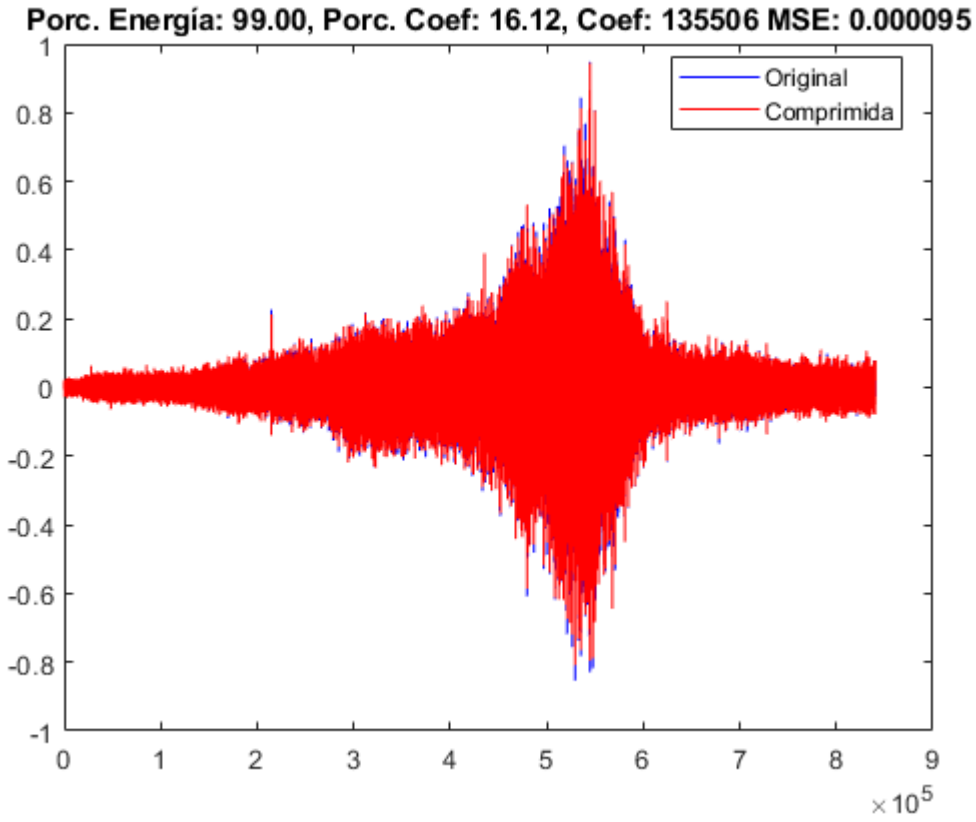
transformada de coseno discreta

```
X = dct(x);

% ordenamos descendentemente los coeficientes encontrados
[XX,ind] = sort(abs(X),'descend');

porcentaje_min = input('Introduce el porcentaje mínimo a retener en la compresión (0.01-
99.99%): ');
while porcentaje_min <= 0 || porcentaje_min >= 100
    porcentaje_min = input('Introduce de nuevo el porcentaje: ');
end

% COMPRESIÓN: número de coeficientes para retener el porcentaje de energía
n = 1;
while norm(X(ind(1:n)))/norm(X) < porcentaje_min/100
   n = n+1;
end
xpc = n/length(X)*100;

% guardamos los coeficientes seleccionados
coef = X(ind(1:n));
% ponemos a cero los coef del resto de la energía
X(ind(n+1:end)) = 0;

% Reconstruimos la señal desde la compresión
xx = idct(X);

% Mostramos la señal original, su reconstrucción y la diferencia entre ambas

figure,plot(1:length(x),x,'b-',1:length(x),xx,'r-')
legend('Original','Comprimida', ...
       'Location','best')
```


Porc. Energía: 50.00, Porc. Coef: 0.26, Coef: 2166 MSE: 0.003597
Original
Comprimida
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6
7
8
9
×10^5


Porc. Energía: 95.00, Porc. Coef: 7.36, Coef: 61838 MSE: 0.000468
Original
Comprimida
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6
7
8
9
×10^5


Porc. Energía: 75.00, Porc. Coef: 1.32, Coef: 11131 MSE: 0.002098
Original
Comprimida
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6
7
8
9
×10^5


Porc. Energía: 99.00, Porc. Coef: 16.12, Coef: 135506 MSE: 0.000095
Original
Comprimida
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
1
2
3
4
5
6
7
8
9
×10^5

Audio de las señales

```
disp('Oir las señales');

% seleccionamos la señal
disp('Indica qué señal deseas oir ');
disp('    1.- Original');
disp('    2.- Reconstruida');
disp('    3.- Diferencia');
op = input('Introduce 1-2-3: ');
switch op
    case 1
        % Escuchar la original
        sound(x,Fs);
    case 2
        % Escuchar la reconstruida (comprimida)
        sound(xx,Fs);
    case 3
        % Escuchar la diferencia
        sound(x-xx,Fs);
    otherwise
        disp('No existe esta opción');
end
```

Published with MATLAB® R2018b