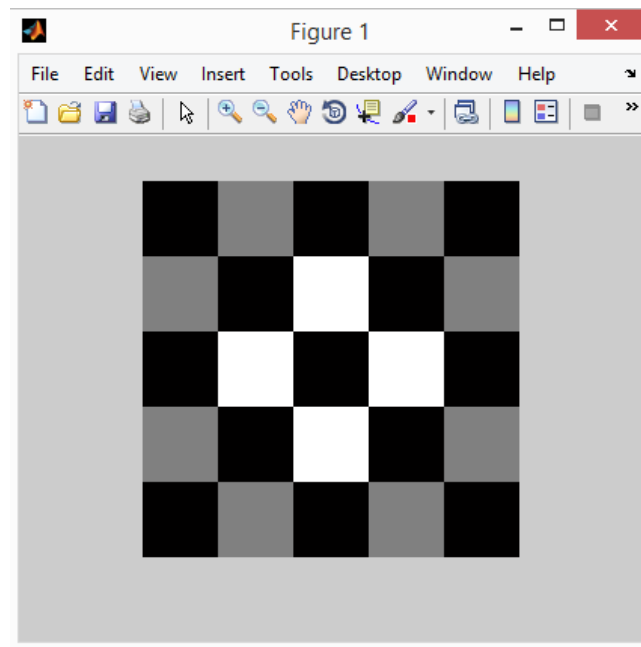


### Ζητούμενα:

1) Για τον υπολογισμό των ζητούμενων δημιουργούμε την συνάρτηση **erwthma1.m** η οποία παίρνει σαν είσοδο την εικόνα που μας ενδιαφέρει και υπολογίζει την εντροπία, την κωδικοποίηση Huffman, πόσα bits/pixel χρησιμοποιούνται για την κωδικοποίηση, ποιος είναι ο λόγος συμπίεσης και ποιος ο πλεονασμός στην αρχική κωδικοποίηση. Τρέχουμε την συνάρτηση και για τις 4 εικόνες και παίρνουμε τα παρακάτω.

chess\_image



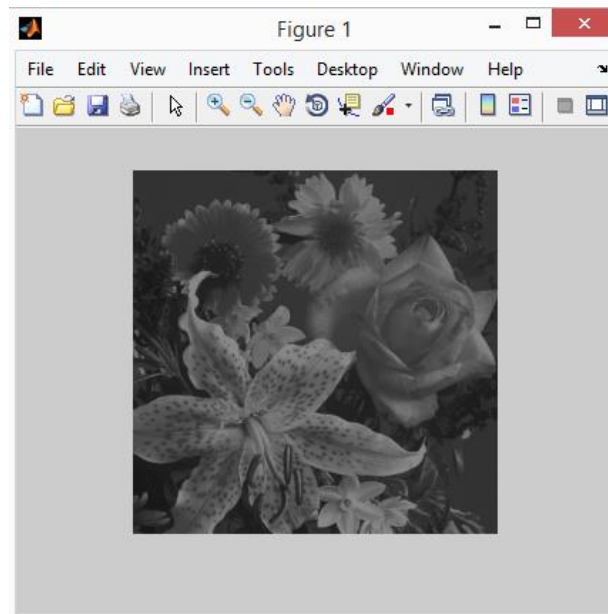
**Entropia:** 1.439628

**Bits ana pixel:** 1.640000

**Logos simpieshs:** 4.878049

**Pleonasmos arikhhs kwdikopoihs:** 0.795000

## flowers



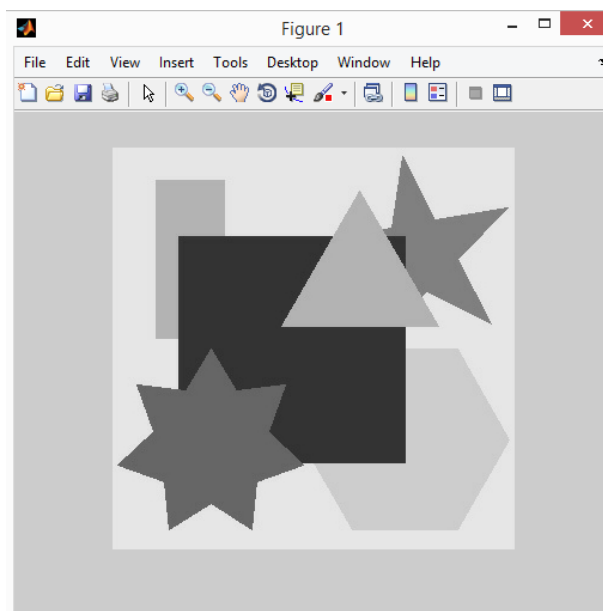
**Entropia:** 6.534571

**Bits ana pixel:** 6.560577

**Logos simpieshs:** 1.219405

**Pleonasmos arxikhs kwdikopoihs:** 0.179928

## geometrical



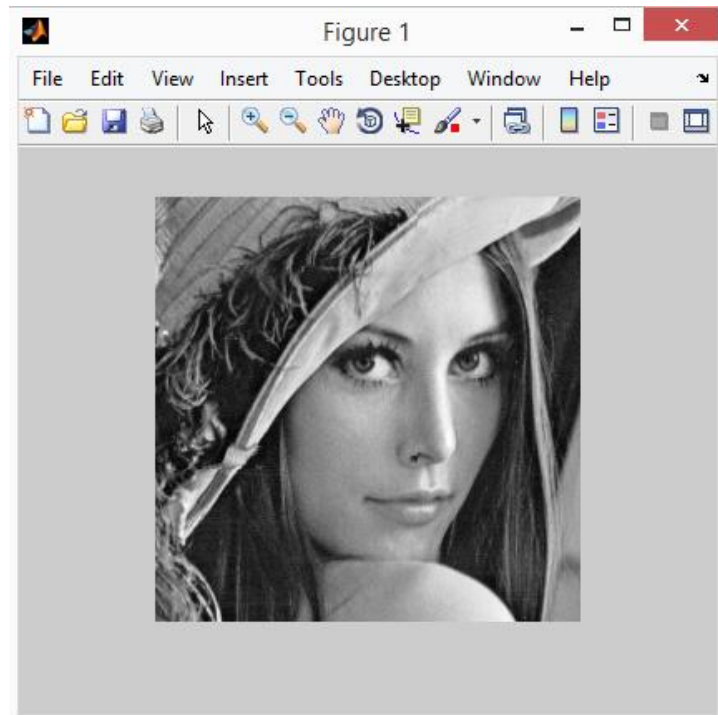
**Entropia:** 3.192038

**Bits ana pixel:** 3.254429

**Logos simpieshs:** 2.458189

**Pleonasmos arxikhs kwdikopoihs:** 0.593196

## lenna



**Entropia:** 7.410816

**Bits ana pixel:** 7.434128

**Logos simpieshs:** 1.076118

**Pleonasmos arxikhs kwdikopoihs:** 0.070734

Η εντροπία σε μια εικόνα μας δίνει τον ελάχιστο αριθμό bits/pixel που χρειαζόμαστε για να κωδικοποιήσουμε την εικόνα. Αν μια εικόνα φέρει περισσότερη πληροφορία σε σχέση με μια άλλη συνεπάγεται πως θα έχει και μεγαλύτερη εντροπία. Για να βρούμε τον αριθμό των bits/pixel που χρησιμοποιούνται διαιρούμε τον συνολικό αριθμό bits της κωδικοποιημένης ακολουθίας με τον συνολικό αριθμό pixels της εικόνας. Αυτός ο αριθμός αντιστοιχίζεται στο μέσο μήκος κώδικα. Ο λόγος συμπίεσης μας δίνει μια αίσθηση για το πόσο έχει μικρύνει η εικόνα μας μετά στην συμπίεση σε σχέση με την αρχική, όσο μεγαλύτερος είναι όμως έχουμε περισσότερες απώλειες. Τέλος ο πλεονασμός μας δίνει το ποσοστό της πλεονάζουσας πληροφορίας. Παρατηρούμε πως όσο πιο λεπτομερής είναι μια εικόνα τόσο πιο δύσκολο είναι να επιτύχουμε μεγαλύτερο λόγο συμπίεσης.

**Ο κώδικας της συνάρτησης φαίνεται παρακάτω:**

```
function [] = erwthma1(img)
    pro=erwthma2_1(img);
    H=0;
    for i=1:256
        %an h pithanothta einai diaforh tou mhdenos
        if(pro(i)~=0)
            H=H+pro(i)*log2(pro(i));
```

```

        end
    end
    H=(-1)*H;
    %ipologismos tou lexikou
    dict=huffmandict((0:255),pro);
    %kwdikopoihsh ths eikonas se mia akolouthia
    enco=huffmanenco(img(:),dict);
    %arithmos bits ana pixel

bits_ana_pixel=size(enco,1)/((size(img,1))*(size(img,2)))
;
    %logos simpieshs c=b/b' b=arithmos bits pou
xreiazontai gia thn
    %anaparastash prin thn simpiesh kai b'=arithmos
bits meta thn simpiesh
    c=((size(img,1))*(size(img,2))*8)/(size(enco,1));
    %ipologismos tou pleonasmou ths simpieshs R=1-(1/c)
    r=1-(1/c);
    fprintf('Entropia: %f\n',H);
    fprintf('Bits ana pixel: %f\n',bits_ana_pixel);
    fprintf('Logos simpieshs: %f\n',c);
    fprintf('Pleonasmos arxikhs kwdikopoihshs:
%f\n',r);
end

```

**2)** Για την υλοποίηση των ζητούμενων δημιουργήσαμε την συνάρτηση **erwthma2.m** η οποία δέχεται σαν είσοδο μία εικόνα και ένα βήμα, το οποίο βήμα μας δηλώνει κατά πόσο θα αυξάνεται το ποσοστό των συντελεστών που θα κρατάμε για κάθε μετασχηματισμό FFT-DCT σε κάθε επανάληψη. Η αρχική τιμή του ποσοστού των συντελεστών που θα κρατάμε αρχίζει από 0% και φτάνει μέχρι το 100%.

**Ο κώδικας μαζί με τα σχόλια που τον εξηγούν φαίνονται παρακάτω:**

```

function [F1,F2,F3,F4,F5,F6,F7,F8] = erwthma2(img,vima)
    img=double(img);
    img=erwthma1_1(img,0,255);
    %-----
-
    %METAVLHTES POY XREIAZONTAI GIA TON FFT
METASXHMATISMO|
    %-----
-
    %ipologismos metasxhmatismou fourier kai shiftarisma
sto kentro
    img_four=fft2(img);
    %ipologismos metrwn apeikonishs ths metasxhmatismou
ths eikonas
    metra_four=abs(img_four);
    %epeidh exoume polles idies times kratame mono tis
monadikes
    metra=unique(metra_four);

```

```

        %sortaroume to logar apo to megalitero sto mikrotero
gia
        %na krathsoume tis megaliteres times
        metra=sort(metra,'descend');
        %trexwn thesh tou pinaka metra
        j=1;
        %ipologismos tou trexontos posostou san arxikh timh
        einai to 0
        sum_pos=0;
        %-----
        %-----
        %METAVLHTES POY XREIAZONTAI GIA TON DCT
        METASXHMATISMO|
        %-----
        -
        %ipologismos metasxhmatismou fourier kai shiftarisma
        sto kentro
        img_dct=dct2(img);
        %ipologismos metrwn apeikonishs ths metasxhmatismou
        ths eikonas
        metra_dct=abs(img_dct);
        %epeidh exoume polles idies times kratame mono tis
        monadikes
        metra2=unique(metra_dct);
        %sortaroume to logar apo to megalitero sto mikrotero
        gia
        %na krathsoume tis megaliteres times
        metra2=sort(metra2,'descend');
        %trexwn thesh tou pinaka metra2
        k=1;
        %ipologismos tou trexontos posostou san arxikh timh
        einai to 0
        sum_pos2=0;
        %-----
        %-----
        %ola ta pososta apo to 0 ews to 100 me to vima pou
        pairnoume san parametro
        pos=(0:vima:100);
        %pinakes gia na kratame to sfalma gia kathe pososto
        sfalma(size(pos,2))=0;
        sfalma2(size(pos,2))=0;
        %diadikasia pou ekteleitai gia kathe pososto
        for i=1:size(pos,2)
            %-----
            %DIADIKASIA IPOLOGISMOU TIMWN GIA TON FFT|
            %-----
            %thelw se kathe neo pososto na mhdenizw alles
            theseis tou pinaka
            %logar ara ton thelw apo thn arxh se kathe for
            temp=img_four;

```

```

        %thelw na krathsw to pos(i) pososto sthn eikona
pou na apoteleitai
        %apo tis megaliteres times tou pinaka logar
        %telika kratame ton deikth j kai xeroume mexri
poia thesh tou pinaka
        %tha krathsoume ta metra
        while sum_pos < pos(i)

sum_pos=sum_pos+(size(find(metra_four==metra(j)),1)*100)/
(size(metra_four,1)*size(metra_four,2));
        %if gia na mhn xeperasoume to pososto pou
theloume na krathsoume
        if (j+1)<size(metra,1)
            %an den xeperasoume to orio pernoume kai
thn epomenh timh
            if
(sum_pos+(size(find(metra_four==metra(j+1)),1)*100)/(size
(metra_four,1)*size(metra_four,2)))<=pos(i)
                j=j+1;
            else
                %alliws stamatame thn while
                sum_pos=pos(i);
            end
        end
    end
    %mhdenizw oles tis theseis tou pinaka temp pou
einai mikrotera ths
    %theshs metra(j)
    temp(find(metra_four<metra(j)))=0;
    %anakatskeuh eikonas me antistrofo metasxhmatismo
fourier
    anakatask=ifft2(temp);
    %gia to parapanw pososto thelw thn logarithmikh
apeionish ths temp
    metasx=log(abs(temp)+1);
    metasx=erwthma1_1(metasx,0,255);
    metasx=(fftshift(metasx));
    %ipologismos ths eikonas sfalmatos apo thn arxikh
afairw thn
    %anakataskevamenh gia trexwn pososto
    eikona_sfalmatos=abs(img-anakatask);

    %ipologismos mesou apolitou sfalmatos exontas to
metro ths diaforas ths arxikh
    %eikonas me thn anakataskevamenh kanoume sum ta
stoixeia auta kai
    %diairoume me to plithos twn pixels

meso_apol_sfalma=sum(eikona_sfalmatos(:))/(size(img,1)*si
ze(img,2));

```

```

%-----
-----
---
%DIADIKASIA IPOLOGISMWN GIA TON DCT|
%-----
%thelw se kathe neo pososto na mhdenizw alles
theseis tou pinaka
%logar ara ton thelw apo thn arxh se kathe for
temp2=img_dct;
%thelw na krathsw to pos(i) pososto sthn eikona
pou na apoteleitai
%apo tis megaliteres times tou pinaka logar
%telika kratame ton deikth j kai xeroume mexri
poia thesh tou pinaka
%tha krathsoume ta metra
while sum_pos2 < pos(i)

sum_pos2=sum_pos2+(size(find(metra_dct==metra2(k)),1)*100
)/(size(metra_dct,1)*size(metra_dct,2));
%if gia na mhn xeperasoume to pososto pou
theloume na krathsoume
if (k+1)<size(metra2,1)
%an den xeperasoume to orio pernoume kai
thn epomenh timh
if
(sum_pos2+(size(find(metra_dct==metra2(k+1)),1)*100)/(siz
e(metra_dct,1)*size(metra_dct,2)))<=pos(i)
k=k+1;
else
%alliws stamatame thn while
sum_pos2=pos(i);
end
end
end
%mhdenizw oles tis theseis tou pinaka temp pou
einai mikrotera ths
%theshs metra(j)
temp2(find(metra_dct<metra2(k)))=0;
%anakatskeuh eikonas me antistrofo metasxhmatismo
fourier
anakatask2=idct2(temp2);
%gia to parapanw pososto thelw thn logarithmikh
apeionish ths temp
metasx2=log(abs(temp2)+1);
metasx2=erwthma1_1(metasx2,0,255);
%ipologismos ths eikonas sfalmatos apo thn arxikh
afairw thn
%anakataskevamenh gia trexwn pososto
eikona_sfalmatos2=abs(img-anakatask2);

```

```

        %ipologismos mesou apolitou sfalmatos exontas to
metro ths diaforas ths arxikhhs
        %eikonas me thn anakataskeuasmenh kanoume sum ta
stoixeia auta kai
        %diairoume me to plithos twn pixels

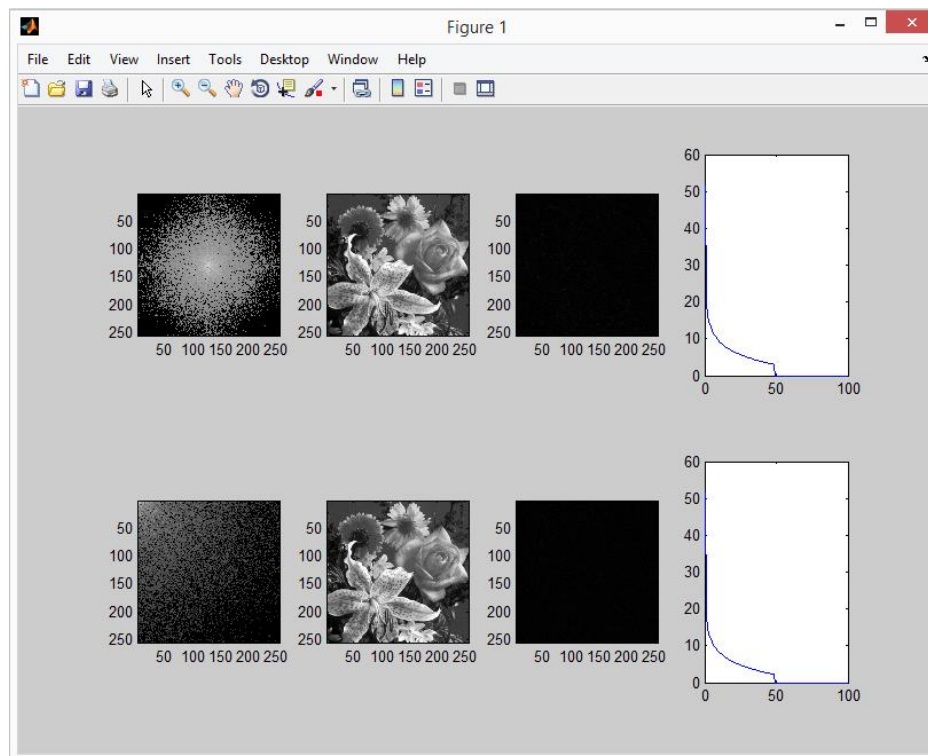
meso_apol_sfalma2=sum(eikona_sfalmatos2(:))/(size(img,1)*
size(img,2));
        %-----
-----
        sfalma(i)=meso_apol_sfalma;
        sfalma2(i)=meso_apol_sfalma2;
        %-----
-----

        %APEIKONISH TOY FRAME|
        %-----
        %APEIKONISH AUTWN POY AFOROUN TON FFT
        %apeikonish metasxhmatismou
        subplot(2,4,1),subimage(uint8(metasx));
        F1(i)=getframe;
        %apeikonish anakataskeuasmenhs eikonas
        subplot(2,4,2),subimage(uint8(anakataask));
        F2(i)=getframe;
        %apeikonish eikonas sfalmatos
        subplot(2,4,3),subimage(uint8(eikona_sfalmatos));
        F3(i)=getframe;
        %apeikonish mesou apolitou sfalmatos
        subplot(2,4,4),plot(pos,sfalma);
        F4(i)=getframe;
        %-----
        %APEIKONISH AUTWN POY AFOROUN TON DCT
        %apeikonish metasxhmatismou
        subplot(2,4,5),subimage(uint8(metasx2));
        F5(i)=getframe;
        %apeikonish anakataskeuasmenhs eikonas
        subplot(2,4,6),subimage(uint8(anakataask2));
        F6(i)=getframe;
        %apeikonish eikonas sfalmatos
        subplot(2,4,7),subimage(uint8(eikona_sfalmatos2));
        F7(i)=getframe;
        %apeikonish mesou apolitou sfalmatos
        subplot(2,4,8),plot(pos,sfalma2);
        F8(i)=getframe;
end
figure;plot(pos,sfalma,pos,sfalma2);
title('MESO APOLITO SFALMA SYNARTHSEI TOY POSOSTOY');
xlabel('Pososto');
ylabel('Meso apolito sfalma');
legend('FFT','DCT');
end

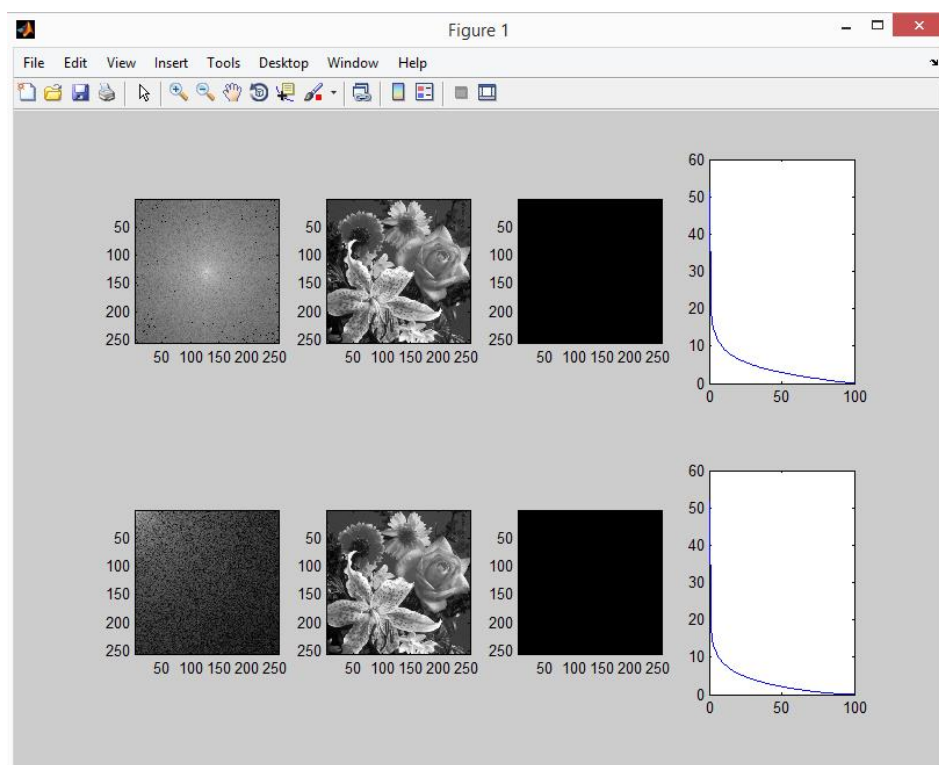
```



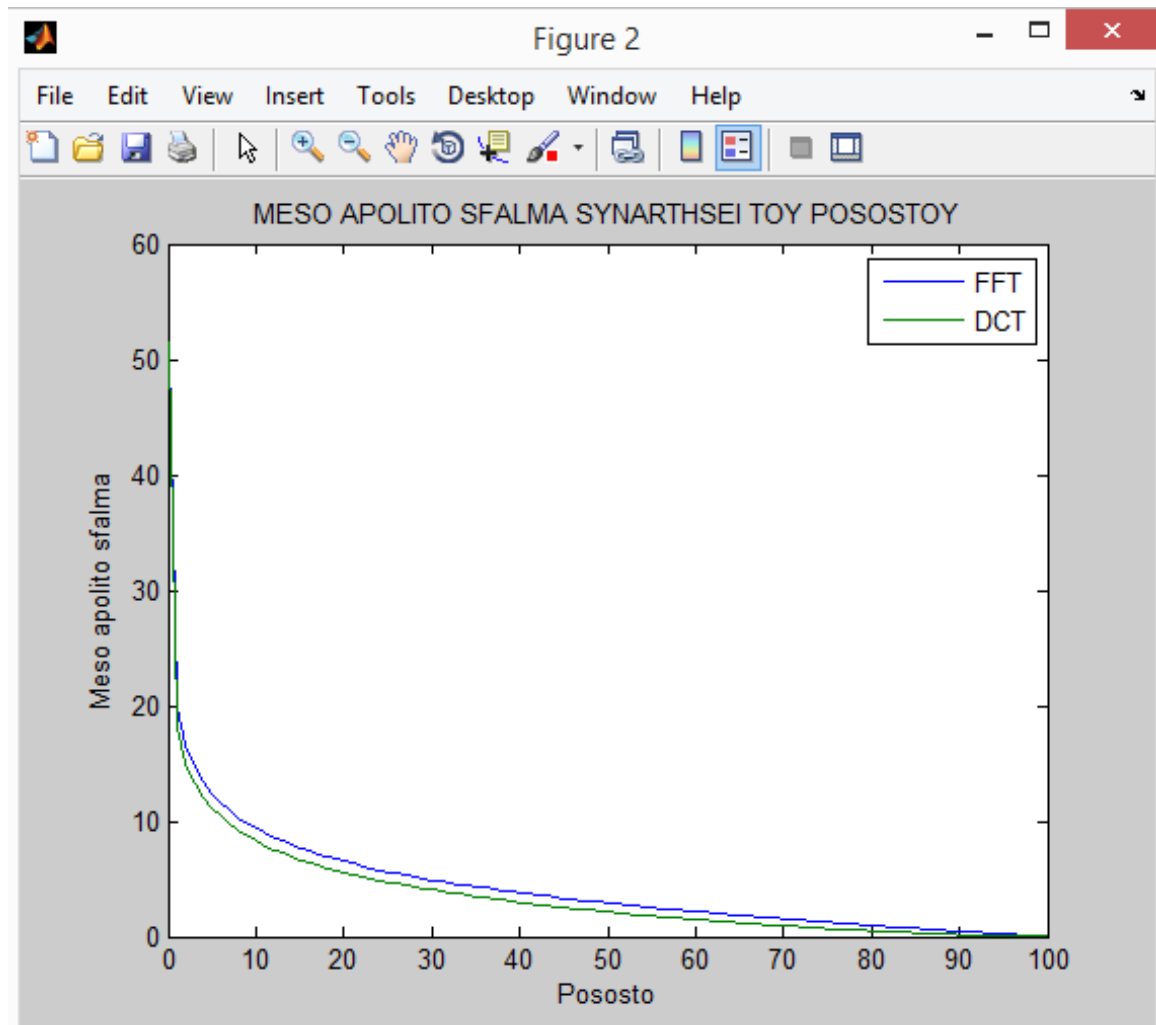
**Εικόνα από την εκτέλεση του κώδικα με ποσοστό συντελεστών που κρατάμε ίσο με 50% για κάθε μετασχηματισμό:**



**Εικόνα από την εκτέλεση του κώδικα με ποσοστό συντελεστών που κρατάμε ίσο με 100% για κάθε μετασχηματισμό:**



Διάγραμμα μέσου απόλυτου σφάλματος (ανακατασκευασμένης εικόνας σε σχέση με την αρχική) προς το ποσοστό των συντελεστών που κρατάμε για κάθε μετασχηματισμό:



Για να συλλέξουμε τα δεδομένα μας εκτελούμε την συνάρτηση γράφοντας **[F1,F2,F3,F4,F5,F6,F7,F8]=erwthma2(x\_fl,1);** , άρα δίνοντας σαν είσοδο την εικόνα και βήμα ίσο με 1. Έπειτα για να σώσουμε τα δεδομένα μας γράφουμε **save fft\_dct\_data** .

Στο παράθυρο των αποτελεσμάτων ότι αφορά τον FFT μετασχηματισμό βρίσκεται στην πρώτη γραμμή, ενώ ότι αφορά τον DCT βρίσκεται στην δεύτερη. Όσον αφορά τις στήλες η πρώτη μας δίνει τον μετασχηματισμό, η δεύτερη ανακατασκευασμένη εικόνα, η τρίτη την απεικόνιση του σφάλματος και η τέταρτη το διάγραμμα του μέσου απόλυτου σφάλματος προς το αντίστοιχο ποσοστό σφάλματος. Η ανανέωση του frame εννοείται πως γίνεται για κάθε διαφορετικό ποσοστό συντελεστών που κρατάμε.