

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ : Τ.Μ.Η/Υ.Π

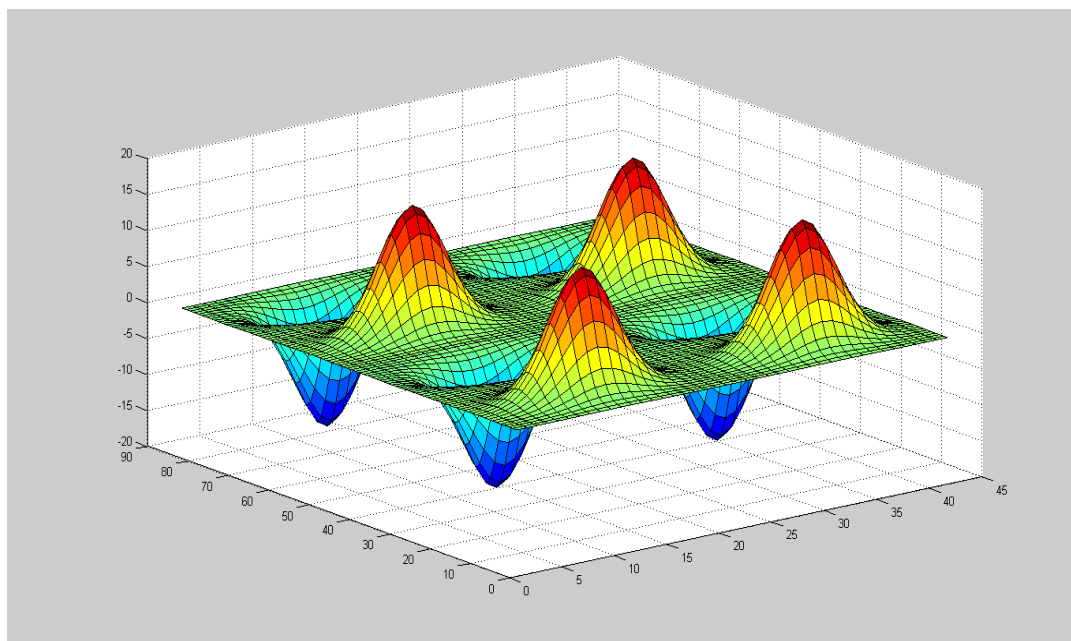
ΕΠΩΝΥΜΟ : ΤΡΙΑΝΤΗΣ

ΟΝΟΜΑ : ΠΑΝΑΓΙΩΤΗΣ

Α.Μ : 5442

ΥΠΟΛΟΓΙΣΤΙΚΗ ΟΡΑΣΗ & ΓΡΑΦΙΚΗ

ΓΚΑΟΥΣΙΑΝΕΣ ΚΑΙ ΛΑΠΛΑΣΙΑΝΕΣ ΠΥΡΑΜΙΔΕΣ



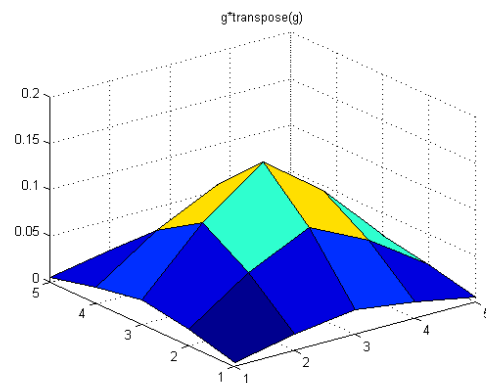
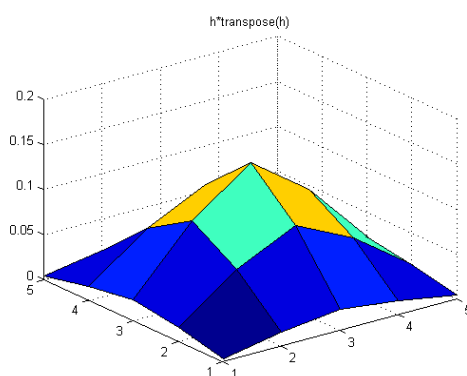
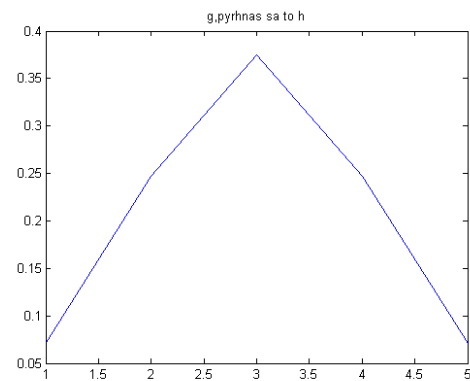
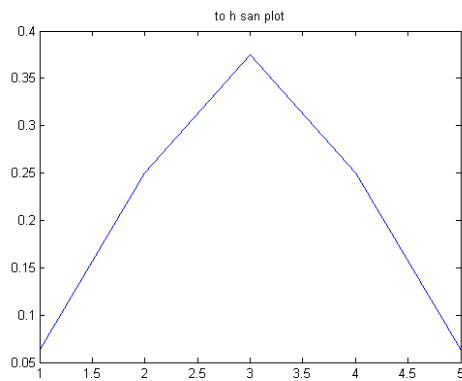
1) Για να αποδείξω ότι το h που μας δόθηκε είναι διακριτό ισοδύναμο γκαουσιανού πυρήνα έκανα το εξής, δημιούργησα ένα γκαουσιανό πυρήνα με διακύμανση $\sigma=1.0638$ η οποία επιλέχθηκε έτσι ώστε για 5 στοιχεία με βάση το τύπο μας δίνει σχεδόν τους αριθμούς του h (1/16,4/16,6/16,4/16,1/16). Ουσιαστικά δημιούργησα το h με το τύπο της κανονικής κατανομής. Υπήρξαν και κάποιες αποκλίσεις βέβαια αλλά ήταν αρκετά μικρές.

Ο τύπος για τη κατανομή :

Τα αποτελέσματα ήταν

h	0,0625	0,25	0,375	0,25	0,0625
g	0,0711	0,2475	0,375	0,2475	0,0711

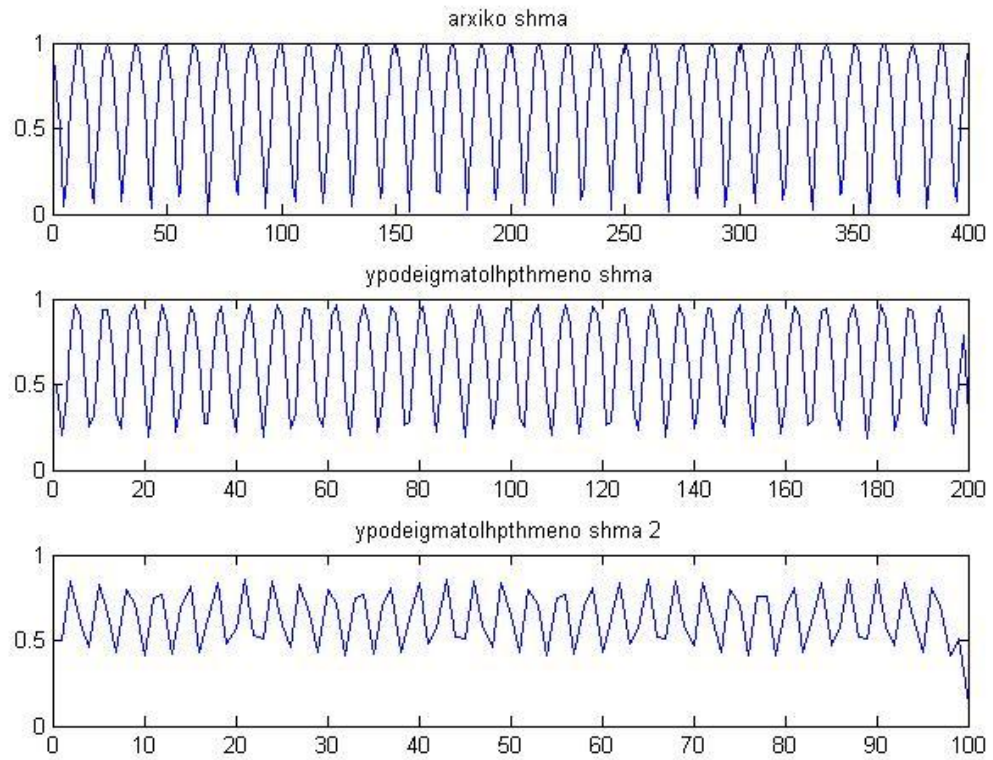
Όπως βλέπουμε υπάρχουν αποκλίσεις αλλά το h πλησιάζει τη συγκεκριμένη κατανομή-πυρήνα. Αυτό φαίνεται και από τις plot & surf (surf πολλαπλασίασα το h με τον ανάστροφο του και το ίδιο και για το g).



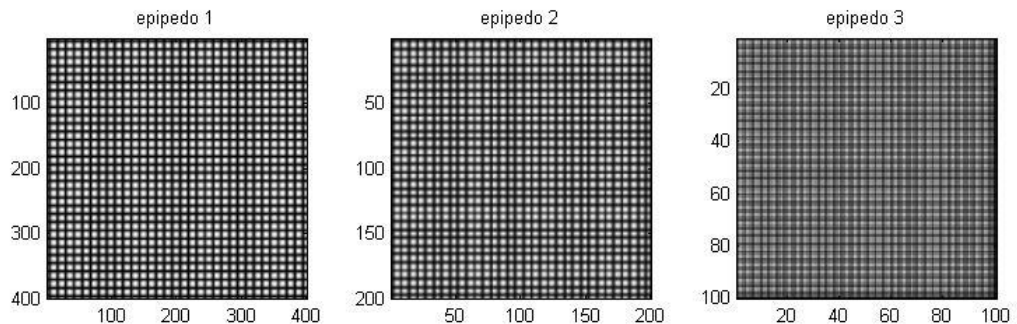
Η δημιουργία των παραπάνω υπάρχει στο script **h_isodynomo_gauss_ker.m** .

2) Για το μονοδιάστατο σήμα που ζητήθηκε πήρα ένα συνημίτονο πάντα θετικό με τιμές έως 1. Στη συνέχεια το φιλτράρισα πολ/ζοντας με το μητρώο Toeplitz και έπειτα το υποδειγματολόγησα με το μητρώο D που δημιουργείται με Kronecker. Με χρήση

plot,imshow(για το σήμα σε δισδιάστατη μορφή) είναι σα πυραμίδα στην εικόνα παρακάτω :



Και σαν εικόνα



Η δημιουργία των παραπάνω υπάρχει στο script **monodiastato_shma.m** .

3) Η επιβεβαίωση ότι από τη πυραμίδα gauss παίρνουμε τη laplace και αντίστροφα (αφού $Laplace_image = normal_image - interpolated_filtered_image \Leftrightarrow$

$interpolated_filtered_image = normal_image - Laplace_image$) γίνεται παρακάτω όπου δημιουργούμε τη συνάρτηση `laplacian_pyramid` που καλεί τη συνάρτηση `gauss_pyramid` και αφαιρεί τα επίπεδα της γκαουσιανής από την αρχική εικόνα και δημιουργεί τη λαπλασιανή πυραμίδα.

4) Όσον αφορά το toolbox :

- **genPyr()** : Η συνάρτηση αυτή δημιουργεί τις ζητούμενες πυραμίδες, Gauss ή Laplace, ανάλογα με το string που θα βάλουμε στην είσοδο μεταβλητής type. Ότι εικόνα δεχθεί σαν είσοδο θα πρέπει πρώτα να έχει γίνει **imread**, στη συνέχεια τη μετατρέπει σε μορφή double. Για να δημιουργήσει τις πυραμίδες η συνάρτηση αυτή στηρίζεται στις συναρτήσεις **pyr_reduce()** & **pyr_expand()**. Παρακάτω θα αναλυθεί τι κάνει η καθεμία. Ως επί των πλείστων στη `gen_pyr()` επιλέγεται ποια πυραμίδα θα δημιουργηθεί.
- **pyr_expand()** : Η `pyr_expand()` υπέρ-τετραπλασιάζει την εικόνα που παίρνει σαν είσοδο. Για να το πετύχει αυτό δημιουργεί ένα kernel με πλάτος 5 και κέντρο βάρος του kernel ίσο με 0.6 , με τη βοήθεια και του γινομένου kronecker. Δημιουργεί 4 υποπυρήνες με τους οποίους επεξεργάζεται την εικόνα. Με βάση αυτούς τους πυρήνες θα γίνει η παρεμβολή και θα προκύψει η νέα εικόνα (μεγαλύτερη). Αρχικά δημιουργώντας μια εικόνα με μηδενικά (αν υποθέσουμε έχει η εικόνα εισόδου M γραμμές και N στήλες , τότε η τελική θα έχει (2M-1,2N-1)) και μετά αφού με κάθε

πυρήνα παρεμβάλλει την αρχική εικόνα την αποθηκεύει στη μεγαλύτερη εικόνα. Εδώ να σημειωθεί πως στην αρχική εικόνα κάνει ένα padding κατά στήλες (βάζει αριστερά και δεξιά ένα αντίγραφο της τελευταίας στήλης, της αριστερής στα αριστερά και της δεξιάς στα δεξιά) και ένα padding κατά γραμμές όπου κάνει το ίδιο πράγμα για τη πρώτη και τη τελευταία γραμμή και τις αποθηκεύει σε 2 ξεχωριστές πλέον εικόνες. Έπειτα αφού υπάρχει το σύνολο των τριών εικόνων (αρχική και οι δύο προσαυξημένες) πλέον παρεμβάλλονται με την εξής μέθοδο, χρησιμοποιείται συνέλιξη και η imfilter. Αποθηκεύεται στη μεγάλη εικόνα με βηματισμό ανά 2 αρχίζοντας από τη 1^η θέση ή τη 2^η για 4 περιπτώσεις από τη πρώτη γραμμή έως το τέλος ανα 2, από τη 2^η γραμμή ανα 2 έως το τέλος και το ίδιο για τις στήλες... αρα συνολικά $2^2 = 4$ περιπτώσεις που είναι η κάθε επεξεργασία των παραπάνω εικόνων με τους 4 διαφορετικούς υποπυρήνες.

- **pyr_reduce()**: Εδώ πάλι χρησιμοποιείται ένας kernel και παράγεται όπως παραπάνω με τα ίδια ακριβώς στοιχεία. Το «φιλτράρισμα» της εικόνας γίνεται από την imfilter εδώ και η τελική εικόνα που δίνεται ως έξοδος είναι το αποτέλεσμα της imfilter υποδειγματοληπτημένο ως εξής, επιλέγοντας ανά δύο τα στοιχεία της εικόνας που έδωσε η imfilter (π.χ 1:2:end,1:2:end). Αυτό να προσθέσουμε ότι γίνεται και για τις 3 διαστάσεις της εικόνας
- **pyrBlend()**: Είναι το script το οποίο ενώνει την εικόνα του μήλου με το πορτοκάλι. Αρχικά δημιουργούνται οι 2 πυραμίδες Laplace για τις εικόνες μας και φτιάχνεται η μάσκα με την οποία θα γίνει η ένωση των εικόνων. Μετά περνιέται η μάσκα και το (1-μάσκα) αντίθετο της μάσκας από φίλτρα θόλωσης (imfilter blurh) και έπειτα για όλα τα επίπεδα της πυραμίδας γίνεται η ένωση με τις μάσκες με τον τύπο (ξεκινώντας εννοείται από το τελευταίο επίπεδο της πυραμίδας):

$$I = img1.*mask + img2.*(1 - mask)$$
Τέλος παρουσιάζεται το αποτέλεσμα της ένωσης του μήλου με το πορτοκάλι με 2 τρόπους, την ένωση που προέκυψε από τη διαδικασία που αναφέρθηκε παραπάνω (ανακατασκευή από τη πυραμίδα Laplace) και την ένωση με τη χρήση των φιλτραρισμένων μασκών που έγινε πιο πάνω.
- **pyrReconstruct()**: Είναι η συνάρτηση η οποία χρησιμοποιεί τη λαπλασιανή πυραμίδα για να ανακατασκευάσει μια εικόνα. Όπως γνωρίζουμε το τελευταίο επίπεδο της πυραμίδας Laplace είναι το ίδιο με της πυραμίδας Gauss έτσι αναδρομικά αν παρεμβάλλουμε την εικόνα του τελευταίου επιπέδου της Laplace πυραμίδας και το προσθέσουμε στο προτελευταίο και το αποτέλεσμα με τη σειρά του στο προηγούμενο κλπ τότε μπορούμε να δημιουργήσουμε την αρχική μας εικόνα.

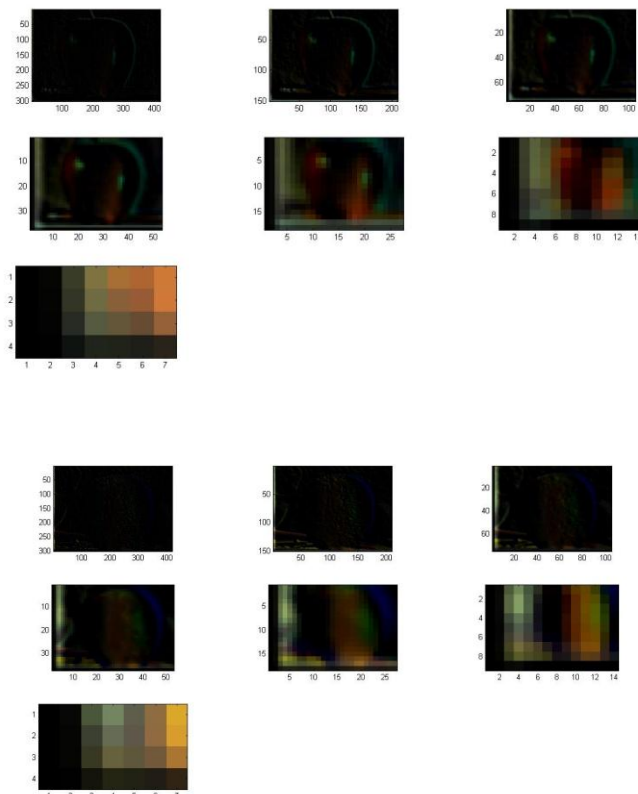
Τώρα για την ένωση-μίξη των εικόνων

Δημιούργησα συναρτήσεις που με βάση τις οδηγίες που δίνονται για το πώς δημιουργούνται οι γκαουσιανες πυραμίδες κάνει ένα φιλτράρισμα στην εικόνα (αναλυτικά σχολιάζεται και ο κώδικας) πολ/ζοντας την εικόνα αριστερά και δεξιά με 2 μητρώα Toeplitz που το ένα είναι ίσο με τις γραμμές τις εικόνες και το άλλο με τις στήλες στη συνέχεια κάνει υποδειγματοληψία και έχει τη δυνατότητα να δείξει τις εικόνες τις πυραμίδας ή όχι και τα αποτελέσματα αποθηκεύονται σε cells. Η συνάρτηση αυτή ονομάζεται `gauss_pyramid()` (όλοι οι κώδικες παρατίθενται στο τέλος). Ακόμη δημιούργησα μία συνάρτηση που από τη συνάρτηση για τις γκαουσιανες δημιουργεί μια λαπλασιανή πυραμίδα `laplacian_pyramid()`. Ακόμη στα πλαίσια των 2 πρώτων ερωτημάτων δημιουργήθηκε και η συνάρτηση μίξης 2 εικόνων από τις λαπλασιανες τους πυραμίδες (`blending2imgs()`).

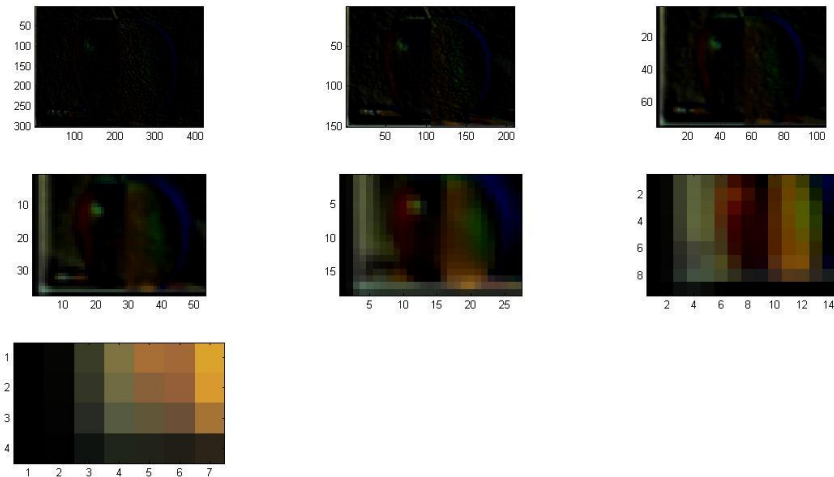
ΜΗΛΟ-ΠΟΡΤΟΚΑΛΙ

1. Δημιουργήθηκε η μάσκα σε ένα script και αποθηκεύτηκε με την `imwrite`.
2. Έγινε χρήση των παραπάνω συναρτήσεων από ένα script (`erotima_1_blending.m`)

Οι πυραμίδες Laplace είναι :



Και ακολουθεί η πυραμίδα ΛΠ για τη μίξη :



Παρατηρούμε ότι μετά από κάποιο σημείο πλέον αρχίζει πάλι η εικόνα να έχει κάποιο χρώμα. Αυτό λογικά γίνεται γιατί δημιουργήθηκαν αρκετά επίπεδα στη πυραμίδα έτσι η διαφορά της γκαουσιανής από τη κανονική σε αυτά τα επίπεδα είναι μικρή οπότε είναι περίπου ίδιο το επίπεδο της λαπλασιανής με τη γκαουσιανή.

Η εικόνα που δημιουργήθηκε από τη μίξη είναι η ακόλουθη :



Βλέπουμε ότι η συραμμένη μέσω των λαπλασιανων πυραμίδων εικόνα στην ένωση έχει κάποια θόλωση η οποία την καθιστά πιο φυσική.

ΧΕΡΙ-MATI

1. Η μάσκα που ορίστηκε έχει ένα ορθογώνιο, στη μέση περίπου της εικόνας της μάσκας, λευκό το οποίο χρησιμοποιείται για να κοπεί το μάτι από την εικόνα με τη γυναίκα. Το 1-μάσκα είναι η μάσκα που πολ/ζεται κόβει το υπόλοιπο χέρι εκτός του ορθογωνίου που θα περιέχει το μάτι και γίνεται η ένωση των εικόνων σε αυτό το σημείο με αυτό τον τρόπο. Η μάσκα έχει δημιουργηθεί με script το οποίο παρατίθεται στο τέλος (**mask_eye.m**).

2. Η μίξη γίνεται και πάλι με τη συνάρτηση **blending2imgs()** που χρησιμοποιεί τι συναρτήσεις που έχω κατασκευάσει. Αν τρέξετε το script που ανέφερα στην οθόνη θα βγουν οι εικόνες από όλα τα στάδια (έχει τεθεί μεταβλητή που μπορεί η παρουσίαση των εικόνων να αποφευχθεί).

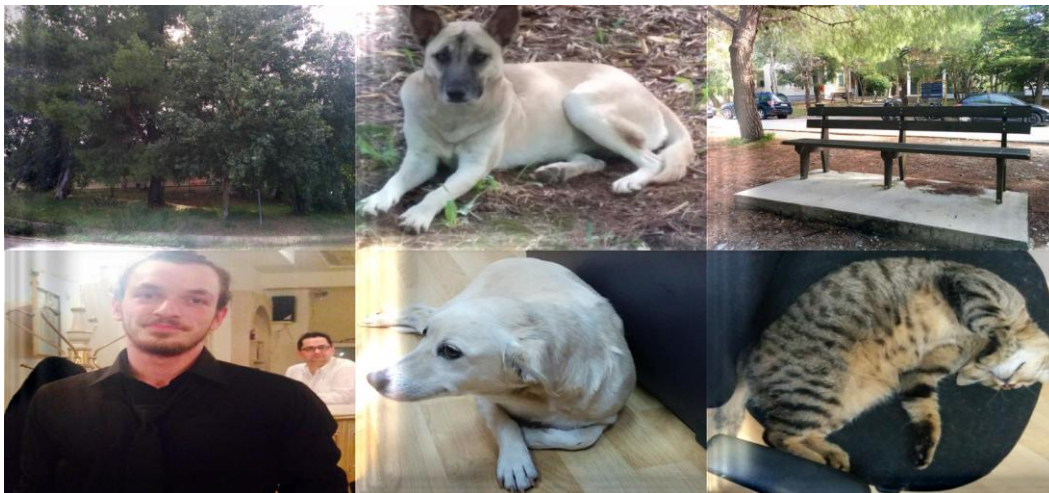
Το αποτέλεσμα της μίξης είναι



ΜΙΞΗ 6 ΕΙΚΟΝΩΝ

1. Η μίξη γι' αυτές τις εικόνες υπάρχει στο script **erotima_3_imgs_to_blend.m**
2. Οι μάσκες δεν αποθηκεύτηκαν γι' αυτό το ερώτημα αλλά υπάρχουν στο script και είναι σχολιασμένες για το πώς επιλέχθηκαν

Εδώ θα παρουσιάσω την εικόνα του αποτελέσματος. Η εικόνα λόγω έντονων χρωματικών διαφορών δεν είχε καλή μίξη. Προσπάθησα λίγο μήπως καταφέρω να πάρω περιγράμματα από τις υψηλές συχνότητες αλλά λόγω χρονικών περιθωρίων και υποχρεώσεων (γενικά του εξαμήνου όχι το συγκεκριμένο μάθημα) δε κατάφερα να το εκμεταλλευτώ.



*** Τέλος να αναφέρω ότι κατά τη διάρκεια του πρότζεκτ πήγαινα και για διάβασμα με το συνάδελφο Γιώργο Μησιακούλη (AM 5355) αλλά δεν υπήρξε συνεργασία για την επίλυση άρα και η επίλυση μας είναι διαφορετική.***

Οι κώδικες που χρησιμοποιήθηκαν είναι :

```
%% apodei3h gia to h
```



```
close all; clear all; clc
```

```
h = 1/16 * [1; 4; 6; 4; 1];  
figure,plot(h),title('to h san plot')
```

```
% orizw mia diakymansh  
s1 = 1.0638;
```

```
% enas counter gia na me voh8hsei na apo8hkeusw se dianysma ta  
apotelesmata  
counter_x = 0;  
for x = -1.94:.97:1.94  
    counter_x = counter_x+1;  
    g(counter_x,1) = (1/(s1*sqrt(2*pi)))*exp(-x^2/(2*s1^2)); %gauss  
me m.t=0  
end  
figure,plot(g),title('g,pyrhnas sa to h')
```

```
% edw 8a kanoume kai surf ta h*h' kai g*g' gia na doume thn omoiothta  
tous  
% kai ws didiastatoi pyrhnes  
figure,surf(h*h'),title('h*transpose(h)')  
figure,surf(g*g'),title('g*transpose(g)')
```

```
%% script gia ena mododiastato shma  
% dhmiourgia enos monodiastatou shmatos to opoio pol/zetai me to  
mhtrwo  
% Toeplitz gia na ginei "filtrarisma" tou shmatos kai na kopoun  
kapoies  
% syxnothtes alla na meinei "idio". gia to logo auto ektos apo tis  
plot pou  
% deixnoun tis times poy pairnei to monodiastato shma to kanw imshow  
afou  
% to pollaplasiasw me anastrofo tou gia na ginei "mhtrwo" kai na exei  
% kapoia morf
```

```
close all; clear all; clc
```

```
for i = 1:400  
y(i,1) = abs(cos(abs(i-401)/4)); % to ebala me abs giati h8ela na  
pairnei >0 times  
end
```

```
h = 1/16 * [1; 4; 6; 4; 1];
```

```
% epipedo 1  
M = length(y);  
e1 = eye(1,M);  
h1 = [h' zeros(1,M-5)];  
T1 = toeplitz(1/16*e1,h1);  
filtered_shma = T1*y;  
z = [1 0];  
D1 = eye(M/2);
```

```

Dkron1 = kron(D1,z);
mikro_shma = Dkron1*filtered_shma;

% epipedo 2
m = length(mikro_shma);
e2 = eye(1,m);
h2 = [h' zeros(1,m-5)];
T2 = toeplitz(1/16*e2,h2);
filtered_2 = T2*mikro_shma;
D2 = eye(m/2);
Dkron2 = kron(D2,z);
mikro_shma_2 = Dkron2*filtered_2;

%subplot - plot
figure,subplot(3,1,1),plot(y),title('arxiko shma')
subplot(3,1,2),plot(mikro_shma),title('ypodeigmatolhpthmeno shma')
subplot(3,1,3),plot(mikro_shma_2),title('ypodeigmatolhpthmeno shma
2')

% subplot - subimage
figure,subplot(1,3,1),subimage(y*y'),title('epipedo 1')
subplot(1,3,2),subimage(mikro_shma*mikro_shma),title('epipedo 2')
subplot(1,3,3),subimage(mikro_shma_2*mikro_shma_2),title('epipedo
3')

```

```

%% function gauss_pyramid
% pairnei mia eikona pou einai se morf h double kai kai analogws me
ton
% ari8mo ,levels, pou dinetai gia to posa epipeda 8eloume na th
kanoume
% efoson to epitrepei kai to fysiko mege8os ths eikonas bebaiws

% h gauss_pyramid apo8hkeuei tis eikones pou h epe3ergasia tous
ginetai apo
% tis parakatw emfwleumenes synarthseis opou epe3hgountai parakatw
pws
% doyleuoun

% img : h eikona mas se morf h img = im2double(imread(.....));
% levels : posa epipeda 8eloume na exei h pyramida, opou to epipedo
mhden
% einai to epipedo ths arxikhs eikonas
% imgs : ena cell pou periexei sth prwth 8esh to mhdeniko epipedo ths
% pyramidas k.o.k
% parousiash : epiloges on || keno || off, gia on || keno tote me
imshow
% parousiazontai ta epipeda ths pyramidas pou dhmiourgeitai
% gia off || keno de parousiazontai ta epipeda ths pyramidas

function [imgs] = gauss_pyramid(img,levels,parousiash)
imgs = cell(levels+1,1);
imgs{1,1} = img;
if levels>1
    fst_lvl = gauss_filtering(img);
    imgs{2,1} = fst_lvl;
    for i = 2:levels
        next_lvls = gauss_filtering(imgs{(i),1});
        imgs{i+1,1} = next_lvls;
    end
end

```

```

end
if levels==1
    one_lvl = gauss_filtering(img);
    imgs{levels+1,1} = one_lvl;
end
if exist('parousiash','var') && strcmp(parousiash,'on')==1
    for j = 1:levels+1
        figure(j),imshow(imgs{j,1})
    end
elseif exist('parousiash','var') && strcmp(parousiash,'off')==1 ||
~exist('parousiash','var')
    disp('oi eikones de parousiazontai')
end
end

%% gauss_filtering
% einai h synarthsh pou pernaei apo filtro tis eikones basei tou
dosmenou
% pyrhna
% me bash to pyrhna kai ta dianysmata e1,e2 dhmiourgountai 2
tetragwnikoi
% pinakes typou Toeplitz o enas (T1) me mege8os oso oi grammes ths
eikonas
% kai o allos (T2) oso oi sthles
% pollaplasiazontas thn eikona apo aristera kai de3ia me autous tous
% pinakes epitygxanetai ena filtrarisma tw n ypshlwn syxnothtw n
function [new_img] = gauss_filtering(img)

[M,N,i] = size(img);
h = 1/16 * [1; 4; 6; 4; 1];
e1 = eye(1,M);
e2 = eye(1,N);
h1 = [h' zeros(1,M-5)];
h2 = [h' zeros(1,N-5)];
T1 = toeplitz(1/16*e1,h1);
T2 = toeplitz(1/16*e2,h2);
%new_img = img;
for j = 1:i
    new_img(:, :, j) = T1*img(:, :, j)*T2;
end
new_img = downscaling(new_img);

%edw para8etw th me8odo me syneli3h tou pyrhna pou do9hke me thn
eikona
% ker = h*h';
% for i = 1:3
%     new_img(:, :, i) = conv2(img(:, :, i), ker, 'same');
% end
% new_img = downscaling(new_img);

end

%% downscaling
% einai h synarthsh pou kanei downscale tis eikones
% opws perigrafhke arxika me th boh9eia tou kronecker kai pali me th
% dhmiourgia 2 pinakwn Dkron1,Dkron2 me mege8h ta misa apo tis
% grammes kai ta misa apo tis sthles antistoixa (mporei ta mege8h na
mhn
% einai akribws sth mesh alla na yparxei mia kanonikopoihsh pros ta
panw)

```

```

% kai ystera me pollaplasiasmo kai pali autwn tw n pinakwn apo
aristera kai
% de3ia me thn eikona pou exoume filtrarei parapanw
% to apotelesma einai na pairnoume th filtrarismenh eikona sto miso
mege8os
% ginetai ypotetraplasiasmos
function [new_level_img] = downscaling(new_img)
[M,N,i] = size(new_img);
z = [1 0];
if rem(M,2)==0
    D1 = eye(M/2);
    Dkron1 = kron(D1,z);
    Dkron1 = Dkron1(1:(M)/2,1:M);
elseif rem(M,2)~=0
    D1 = eye((M+1)/2);
    Dkron1 = kron(D1,z);
    Dkron1 = Dkron1(1:(M-1)/2,1:M);
end
if rem(N,2)==0
    D2 = eye(N/2);
    Dkron2 = kron(D2,z');
elseif rem(N,2)~=0
    D2 = eye((N+1)/2);
    Dkron2 = kron(D2,z');
    Dkron2 = Dkron2(1:N,1:(N+1)/2);
end

for j = 1:i
    new_level_img(:, :, j) = Dkron1*new_img(:, :, j)*Dkron2;
end
end

```

```

%% function laplace_pyramid
% h laplace pyramida tw n eikonwn
% img
% parousiash : epiloges on || keno || off. gia on tote me imshow
% parousiazontai ta epipeda ths pyramidas pou dhmiourgeitai
% gia off || keno de parousiazontai ta epipeda ths pyramidas

function [gauss_imgs, laplace_imgs] =
laplacian_pyramid(img,levels,parousiash)
% arxika kaleitai h gauss_pyramid kai epeita me bash th 8ewria
% dhmiourgeitai h pyramida laplace
laplace_imgs = cell(levels+1,1);
if exist('parousiash','var')
    gauss_imgs = gauss_pyramid(img,levels,parousiash);
else gauss_imgs = gauss_pyramid(img,levels);
end
for i = 1:levels+1
    if i<levels+1
        temp_img = gauss_imgs{i,1}-imresize(gauss_imgs{i+1,1},...
            [size((gauss_imgs{i,1}),1),size((gauss_imgs{i,1}),2)]);
        laplace_imgs{i,1} = temp_img;
    elseif i==levels+1
        laplace_imgs{i,1} = gauss_imgs{i,1};
    end
end
end
if exist('parousiash','var') && strcmp(parousiash,'on')==1
    for j = 1:levels+1

```

```

        i = levels+1;
        figure(i+j),imshow(laplace_imgs{j,1})
    end
elseif exist('parousiash','var') && strcmp(parousiash,'off')==1 ||
~exist('parousiash','var')
    disp('oi eikones de parousiazontai')
end
end
%% synarthsh blending2imgs
% h synarthsh auth dhmiourgh8hke gia na kanei blend se 2 eikones
% afou dw8ei h maska kai oi eikones pou 8eloume mporoume na
epile3oume kai
% posa epipeda pyramidas 8a kanei kai sth synexeia kalei tis
synarthseis
% gia na kanei th mi3h
% mask : h maska me thn opoia 8a syn8esoume thn eikona pou 8eloume
% img_a , img_b : oi eikones pou 8eloume na syn8esoume
% levels : ta epipeda pyramidwn pou 8eloume na dhmiourghsoume
% parousiash : epiloges on || keno || off, gia on || keno tote me
imshow
% parousiazontai ta epipeda ths pyramidas pou dhmiourgeitai
% gia off || keno de parousiazontai ta epipeda ths pyramidas
% blend : h laplacianh pyramida me mi3h
function [blend,blended_img] =
blending2imgs(mask,img_a,img_b,levels,parousiash)
[xa,ya,~] = size(img_a);
[xb,yb,~] = size(img_b);

% edw apofasizoume poio 8a einai to mege8os ths eikonas pou 8a
dhmiourgh8ei
% (exw epile3ei na pairnei sa mege8os th megalyterh eikona)
% kai sth synexeia me thn imresize tis kanoume oles to idio mege8os
if xa>xb
    M = xa;
elseif xa==xb
    M = xa;
else M = xb;
end

if ya>yb
    N = ya;
elseif ya==yb
    N = ya;
else N = yb;
end

mask = imresize(mask,[M N]);
img_a = imresize(img_a,[M N]);
img_b = imresize(img_b,[M N]);

% kaloume tis synarthseis gauss_pyramid gia th maska kai
laplacian_pyramid
% gia tis eikones pou 8a syn8esoume
if exist('parousiash','var')
    [gauss_mask] = gauss_pyramid(mask,levels,parousiash);
    [gauss_img_a, laplace_img_a] =
laplacian_pyramid(img_a,levels,parousiash);
    [gauss_img_b, laplace_img_b] =
laplacian_pyramid(img_b,levels,parousiash);
else [gauss_mask] = gauss_pyramid(mask,levels);

```

```

        [gauss_img_a, laplace_img_a] = laplacian_pyramid(img_a, levels);
        [gauss_img_b, laplace_img_b] = laplacian_pyramid(img_b, levels);
    end

    blend = cell([levels+1,1]);

    % edw pollaplasiazontas ta epipeda ths maskas me ta epipeda tw n
    laplacianwn
    % pyramidwn tw n eikonwn kai ta pros8etoume gia na ginei h syn8esh
    for j = 1:levels+1
        blend{j,1} = (1-gauss_mask{j,1}).*laplace_img_a{j,1} +
        gauss_mask{j,1}.*laplace_img_b{j,1};
    end

    % 8etoume to teleutaio epipedo ths mi3hs pou egine pio panw san
    blended_img
    blended_img = blend{end,1};

    % edw pros8etoume to blended_img me ta prohgoumena epipeda tou,
    anadromika
    % dhldh to megalwnoume kai to pros8etoume sto prohgoumeno, meta to
    % apotelesma ths pros8eshs me to prohgoumeno akomh ews otou ftasoume
    sto
    % epipedo 0 , dhldh to epipedo ths arxikhs eikonas (ths mi3hs
    dhldh)
    for i = (levels):-1:1
        [m,n,~] = size(blend{i,1});
        blended_img = imresize(blended_img, [m n]);
        blended_img = blended_img + blend{i,1};
    end
    if exist('parousiash','var') && strcmp(parousiash,'on')==1
        figure(2*levels+1), imshow(blended_img)
    elseif exist('parousiash','var') && strcmp(parousiash,'off')==1 ||
    ~exist('parousiash','var')
        disp('oi eikones de parousiazontai')
    end
end



---



%% script gia th dhmiourgia maskas gia to mhlo kai to portokali

close all; clear all; clc

mask_ap_or = zeros(301,420,3);

for i = 210:420
    for j = 1:3
        mask_ap_or(:,i,j) = 1;
    end
end

% imwrite(mask_ap_or,'mask_ap_or.png');



---



%% erotima 1 kanei blend tis eikones apple.jpg & orange.jpg

mask_ap_or = im2double(imread('mask_ap_or.png'));
apple = im2double(imread('apple.jpg'));
orange = im2double(imread('orange.jpg'));

```

```

[pyramida_mi3hs,mhlo_portokali] =
blending2imgs(mask_ap_or,apple,orange,6,'on');
[~,apple_lp_pyr] = laplacian_pyramid(apple,6);
[~,orange_lp_pyr] = laplacian_pyramid(orange,6);
% to kommati auto htan gia na bgalw tis eikones se ena figure gia ola
ta
% epipeda (shmeiwsh oti to mikrotera epipeda fainetai me tatragwnakia
giati
% einai poly mikra se mege8os ,ta3hs 7x13....

```

```

% figure
% for i = 1:7
%     subplot(3,3,i),subimage(pyramida_mi3hs{i,1});
% end
% figure
% for i = 1:7
%     subplot(3,3,i),subimage(apple_lp_pyr{i,1})
% end
% figure
% for i = 1:7
%     subplot(3,3,i),subimage(orange_lp_pyr{i,1})
% end

```

```

%% script pou dhmiourgei th maska gia thn enwsh tou matiou me to xeri

```

```

close all; clear all; clc

```

```

mask_eye = zeros(190,195,3);
for v = 73:140
    for y = 82:115
        for i = 1:3
            mask_eye(y,v,i) = 1;
        end
    end
end

```

```

%imwrite(mask_eye,'mask_eye.png');

```

```

%% erotima 2 kanei blend tis eikones woman.png & hand.png

```

```

mask_eye = im2double(imread('mask_eye.png'));
hand = im2double(imread('hand.png'));
woman = im2double(imread('woman.png'));

```

```

[pyr_mi3hs_lp,xeri_mati] = blending2imgs(mask_eye,hand,woman,5,'on');

```

```

%% script pou kanei mi3h tis 6 eikones

```

```

close all; clear all; clc

```

```

% arxika diabazoume tis eikones
p200_ar = im2double(imread('P200.jpg'));
bench_ar = im2double(imread('bench.jpg'));

```

```

dog = im2double(imread('dog1.jpg'));
dogs = im2double(imread('dog2.jpg'));
meow = im2double(imread('cat.jpg'));
mee = im2double(imread('me.jpg'));

% edw kanoume ena eidous crop epilegontas ta pixels apo tis eikones
pou
% 8eloume na kanoume mi3h
dog1(1:1100,1:1300,1:3) = dog(701:1800,1201:2500,1:3);
dog2(1:1100,1:1300,1:3) = dogs(701:1800,1201:2500,1:3);
cat(1:1300,1:1600,1:3) = meow(601:1900,1001:2600,1:3);
me(1:900,1:800,1:3) = mee(251:1150,301:1100,1:3);

% edw ginetai mia smikrynsh ws ekshs
% epilegetai to fonto ths mi3hs na einai oso oi arxikes eikones kai
kobetai
% se 6 kommatia sta opoia 8a topo8ethsoume tis eikones mas etsi
exoume ta
% e3hs mege8h eikonwn (408,544)
p200 = imresize(p200_ar,[408 544]);
bench = imresize(bench_ar,[408 544]);
dog1 = imresize(dog1,[408 544]);
dog2 = imresize(dog2,[408 544]);
cat = imresize(cat,[408 544]);
me = imresize(me,[408 544]);

% edw pleon epilegoume mesa sto megalo mas fonto th 8esh pou 8a exei
h ka8e
% eikona kai meta kanoume padding me mhdenika etsi exoume 6 eikones
oi
% opoies periexoun th ka8e eikona ths mi3hs sto epi8ymhto tetragwno
p200_full = [p200 zeros(408,544,3) zeros(408,544,3);
zeros(408,544,3) zeros(408,544,3) zeros(408,544,3)];
dog1_full = [zeros(408,544,3) dog1 zeros(408,544,3);
zeros(408,544,3) zeros(408,544,3) zeros(408,544,3)];
bench_full = [zeros(408,544,3) zeros(408,544,3) bench;
zeros(408,544,3) zeros(408,544,3) zeros(408,544,3)];
cat_full = [zeros(408,544,3) zeros(408,544,3) zeros(408,544,3);
zeros(408,544,3) zeros(408,544,3) cat];
dog2_full = [zeros(408,544,3) zeros(408,544,3) zeros(408,544,3);
zeros(408,544,3) dog2 zeros(408,544,3)];
me_full = [zeros(408,544,3) zeros(408,544,3) zeros(408,544,3);
me zeros(408,544,3) zeros(408,544,3)];

% gia tis maskes isxyei oti isxyei kai pio panw gia tis eikones mas
mono
% pou sthn epi8ymhth 8esh ka8e maskas exoume 1 (dhladh ekei p
brisketai h
% eikona einai aspro) kai to ypoloipo mauro
mask_p200 = [ones(408,544,3) zeros(408,544,3) zeros(408,544,3);
zeros(408,544,3) zeros(408,544,3) zeros(408,544,3)];
mask_dog1 = [zeros(408,544,3) ones(408,544,3) zeros(408,544,3);
zeros(408,544,3) zeros(408,544,3) zeros(408,544,3)];
mask_bench = [zeros(408,544,3) zeros(408,544,3) ones(408,544,3);
zeros(408,544,3) zeros(408,544,3) zeros(408,544,3)];
mask_cat = [zeros(408,544,3) zeros(408,544,3) zeros(408,544,3);
zeros(408,544,3) zeros(408,544,3) ones(408,544,3)];

```



```

mask_dog2 = [zeros(408,544,3) zeros(408,544,3) zeros(408,544,3);
             zeros(408,544,3) ones(408,544,3) zeros(408,544,3)];
mask_me = [zeros(408,544,3) zeros(408,544,3) zeros(408,544,3);
           ones(408,544,3) zeros(408,544,3) zeros(408,544,3)];

% epeita kata ta gnwsta pairnoume gauss gia tis maskes
gauss_mask_p200 = gauss_pyramid(mask_p200,5,'off');
gauss_mask_dog1 = gauss_pyramid(mask_dog1,5,'off');
gauss_mask_bench = gauss_pyramid(mask_bench,5,'off');
gauss_mask_cat = gauss_pyramid(mask_cat,5,'off');
gauss_mask_dog2 = gauss_pyramid(mask_dog2,5,'off');
gauss_mask_me = gauss_pyramid(mask_me,5,'off');

% laplace gia tis eikones
[~, laplace_p200] = laplacian_pyramid(p200_full,5,'off');
[~, laplace_dog1] = laplacian_pyramid(dog1_full,5,'off');
[~, laplace_bench] = laplacian_pyramid(bench_full,5,'off');
[~, laplace_cat] = laplacian_pyramid(cat_full,5,'off');
[~, laplace_dog2] = laplacian_pyramid(dog2_full,5,'off');
[~, laplace_me] = laplacian_pyramid(me_full,5,'off');

% edw ginetai h mi3h tw n laplasianwn eikonwn pros8etontas ta epipeda
apo
% tis maskes pollaplasiasmenes me ta antistoixa epipeda tw n
laplasianwn
% eikonwn
for j = 1:6
    blend{j,1} = laplace_p200{j,1}.*gauss_mask_p200{j,1}...
        + laplace_dog1{j,1}.*gauss_mask_dog1{j,1}...
        + laplace_bench{j,1}.*gauss_mask_bench{j,1}...
        + laplace_cat{j,1}.*gauss_mask_cat{j,1} +...
        + laplace_dog2{j,1}.*gauss_mask_dog2{j,1}...
        + laplace_me{j,1}.*gauss_mask_me{j,1};
end

% pairnoume to teleutaio epipedo kai paremballontas to to pros8etoume
sta
% epomena epipeda (dld ta pio xamhla) me th gnwrimh anadromikh sxesh
blended_img = blend(end,1);

for i = 5:-1:1
    [m,n,~] = size(blend{i,1});
    blended_img = imresize(blended_img, [m n]);
    blended_img = blended_img + blend{i,1};
end

imshow(blended_img)

```