



**Πανεπιστήμιο Αιγαίου**

**Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών  
Συστημάτων**

**321 - 6555 Πολυμέσα**

Διδάσκων: Καρύμπαλη Ειρήνη

---

## **1<sup>η</sup> Άσκηση**

---

Εργαστηριακοί Συνεργάτες: Καρύμπαλη Ειρήνη

Ευκαρπίδης Κωνσταντίνος

icsd15051

Κωστούλας Δημήτριος

icsd13095

Σάμος, 2/12, 2018



## Περιεχόμενα

<b>1 ΖΗΤΗΜΑ 1<sup>ο</sup></b>	<b>3</b>
1.1 ΑΠΕΙΚΟΝΙΣΗ ΚΑΝΑΛΙΩΝ ΉΧΟΥ ΣΕ FIGURE	3
1.2 ΕΝΑΛΛΑΓΕΣ ΣΥΧΝΟΤΗΤΑΣ ΔΕΙΓΜΑΤΟΛΗΨΙΑΣ	3
1.3 ΥΠΟΔΕΙΓΜΑΤΟΛΗΨΙΑ ΑΡΧΕΙΟΥ ΉΧΟΥ	4
1.4 ΣΥΝΑΡΤΗΣΗ ΕΞΑΣΘΕΝΙΣΗΣ (FADE-OUT)	4
1.5 ΣΥΝΑΡΤΗΣΗ ΑΥΞΗΣΗΣ ΈΝΤΑΣΗΣ (FADE-IN)	5
1.6 ΣΥΝΑΡΤΗΣΗ ΠΡΟΣΘΗΚΗΣ ΛΕΥΚΟΥ ΘΟΡΥΒΟΥ GAUSSIAN	5
1.7 ΑΡΧΕΙΟ SUBJECT1.MAT	7
<b>2 ΖΗΤΗΜΑ 2<sup>ο</sup></b>	<b>8</b>
2.1 ΣΥΝΑΡΤΗΣΗ ΔΗΜΙΟΥΡΓΙΑΣ ΕΙΚΟΝΑΣ ΜΕ ΟΜΟΚΕΝΤΡΟΥΣ ΚΥΚΛΟΥΣ	8
2.2 ΥΠΟΔΕΙΓΜΑΤΟΛΗΨΙΑ ΕΙΚΟΝΑΣ	10
2.3 ΣΥΝΑΡΤΗΣΗ ΕΞΑΛΕΙΨΗΣ ΠΑΡΑΜΟΡΦΩΣΗΣ	10
2.4 ΑΡΧΕΙΟ SUBJECT2.MAT	12



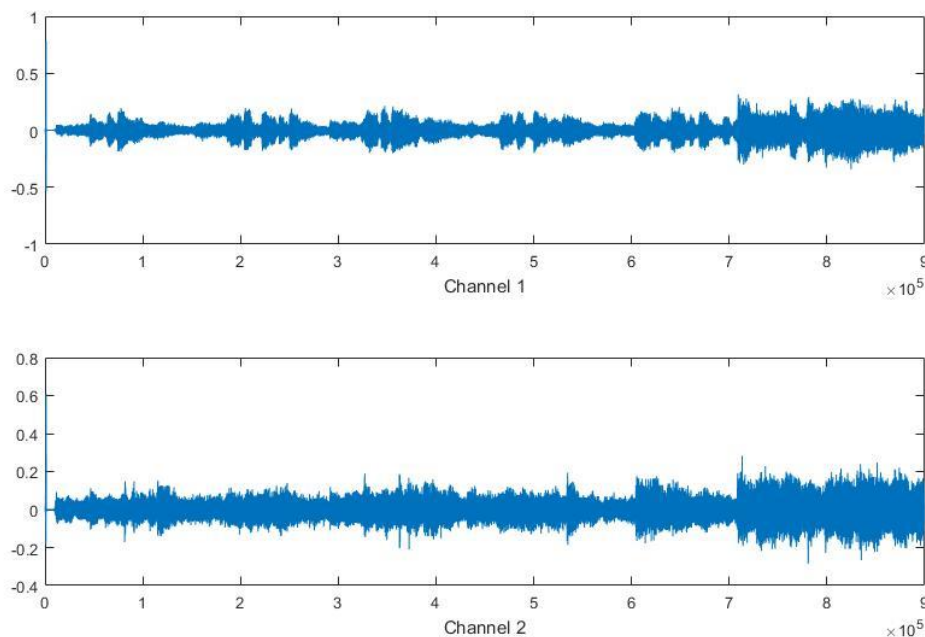
## 1 Ζήτημα 1<sup>ο</sup>

Στο πρώτο ζήτημα μας δίνεται για επεξεργασία ένα αρχείο με όνομα “soundtest.mat”. Στην ουσία πρόκειται για ένα αρχείο ήχου , που είναι ένας πίνακας  $900000 \times 2$  , και μία μεταβλητή που αναπαριστά την συχνότητα δειγματοληψίας.

### 1.1 Απεικόνιση καναλιών ήχου σε figure

Για την απεικόνιση και των δύο καναλιών σε ένα figure τρέξαμε την εντολή :

“figure; subplot (2,1,1); plot(x(:,1)); subplot (2,1,2); plot(x(:,2));”



Εικόνα 1.1.1 : Απεικόνιση και των 2 καναλιών σε 1 figure

### 1.2 Εναλλαγές συχνότητας δειγματοληψίας

Στο ερώτημα αυτό μας ζητείται να κάνουμε χρήση της συνάρτησης sound , ώστε να ακούσουμε το αρχείο ήχου που μας επισυνάφτηκε. Την μία φορά διπλασιάζουμε την συχνότητα δειγματοληψίας “sound(x,2\*fs)” και παρατηρούμε ότι το κομμάτι πηγαίνει πολύ πιο γρήγορα και για την ακρίβεια 2 φορές . Την άλλη φορά υποδιπλασιάσαμε την συχνότητα “sound (x, fs/2)” και παρατηρήσαμε ότι το κομμάτι ακούγεται πολύ αργό.



### 1.3 Υποδειγματοληψία αρχείου ήχου

Για να πετύχουμε υποδειγματοληψία για διάφορες τιμές χρησιμοποιούμε την συνάρτηση `sound` απλά πρέπει να ορίσουμε το βήμα. Δηλαδή “`sound (x(1:n:900000),fs/n)`”, με  $n \geq 1$ . Με αυτό τον τρόπο του λέμε στην ουσία αντί να διαβάζεις όλο τον πίνακα να διαβάζεις με ένα συγκεκριμένο βήμα. Όσο μεγαλώνει το  $n$ , τόσο περισσότερη αλλοίωση παρατηρούμε. Όπως καταλαβαίνουμε το τελικό αποτέλεσμα σε καμία περίπτωση δεν μπορεί να είναι το ίδιο, αφού χάνεται πληροφορία.

### 1.4 Συνάρτηση εξασθένησης (Fade-out)

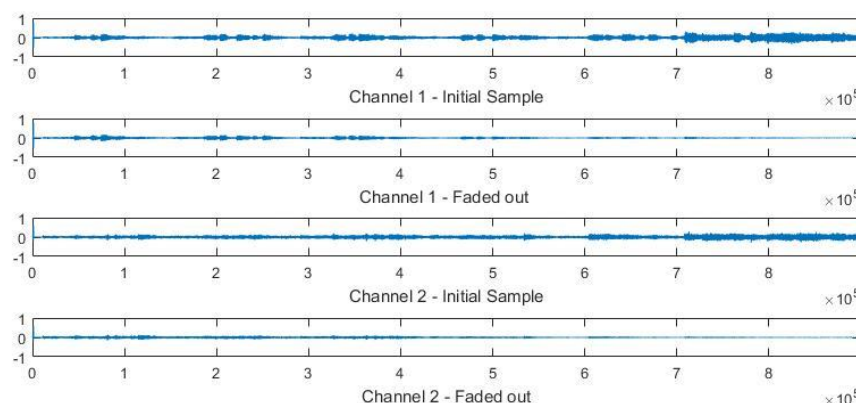
Για την σταδιακή μείωση της έντασης του ήχου έπρεπε να φτιάξουμε μία συνάρτηση η οποία θα μειώνει το πλάτος του σήματος γραμμικά. Δηλαδή αν για πλάτος 1 έχουμε την μέγιστη ένταση έπρεπε να μειώνεται με βήμα όσο και στάθμες του αρχείου, 900000. Άρα έπρεπε να πολλαπλασιάσουμε την ένδειξη κάθε στάθμης ξεκινώντας από 1 με βήμα αρνητικό  $1/900000$  φτάνοντας στο 0 όπου και θα τελείωνε το κομμάτι.

```
1 function [ x ] = fadeout(x)
2
3 for j = 1:2
4     counter=900000;
5     for i = 1:900000
6         x(i,j) = x(i,j) * (counter/900000); %Γραμμική μείωση του πλάτους
7         counter=counter-1; %λόγω μείωσης του μετρητή
8     end %initial_value=1 , final_value=0
9 end
10
11 end
```

Εικόνα 1.4.1 : Συνάρτηση γραμμικής εξασθένησης έντασης ήχου

Για να είναι εμφανής λοιπόν η αναπαράσταση στο figure βάλουμε πρώτα το 1<sup>ο</sup> κανάλι του αναλλοίωτου αρχείου και έπειτα το 1<sup>ο</sup> κανάλι του εξασθενημένου αρχείου. Αντίστοιχα κάναμε το ίδιο και με το 2<sup>ο</sup> κανάλι.

```
figure; subplot (4,1,1); plot(x(:,1)); subplot(4,1,2); plot(y(:,1)); subplot(4,1,3); plot(x(:,2));
subplot(4,1,4); plot(y(:,2));
```



Εικόνα 1.4.2 : Γραφική αναπαράσταση των καναλιών πριν & μετά την εξασθένηση



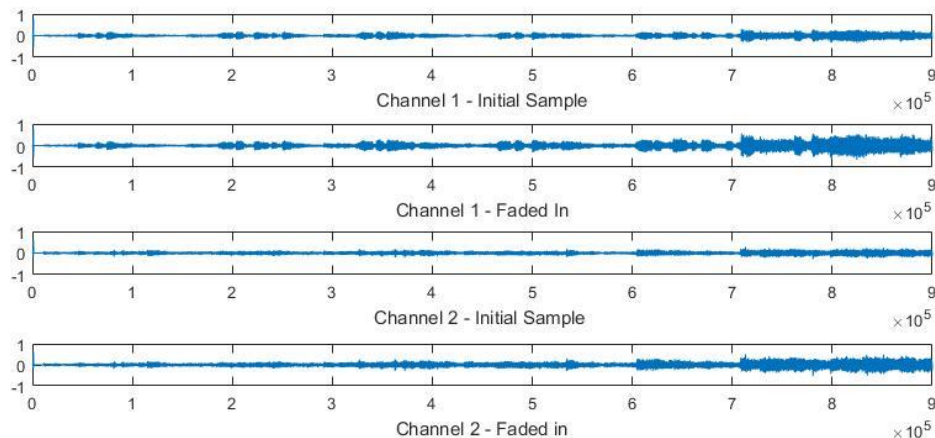
## 1.5 Συνάρτηση αύξησης έντασης (Fade-in)

Για την ισχυροποίηση της έντασης (πλάτος) του ήχου έπρεπε να ακολουθήσουμε την ίδια διαδικασία με πριν, με την διαφορά ότι αντί να μειώνουμε σταδιακά από το 1 μέχρι το 0, τώρα έπρεπε να αυξάνουμε το πλάτος από το 1 στο 2 για παράδειγμα.

```
1 function [ x ] = fadein( x )
2
3 for j = 1:2
4     counter=900000;
5     for i = 1:900000
6         x(i,j) = x(i,j) * (counter/900000); %Γραμμική αύξηση του πλάτους
7         counter=counter+1; %λόγω αύξησης του μετρητή
8     end %initial_value=1 , final_value=2
9 end
10
11 end
```

Εικόνα 1.5.1 : Συνάρτηση γραμμικής αύξησης έντασης ήχου

Για την απεικόνιση των γραφικών παραστάσεων χρησιμοποιήσαμε την ίδια λογική όπως και στην υποενότητα 1.4.



Εικόνα 1.5.2 : Γραφική αναπαράσταση των καναλιών πριν & μετά την αύξηση

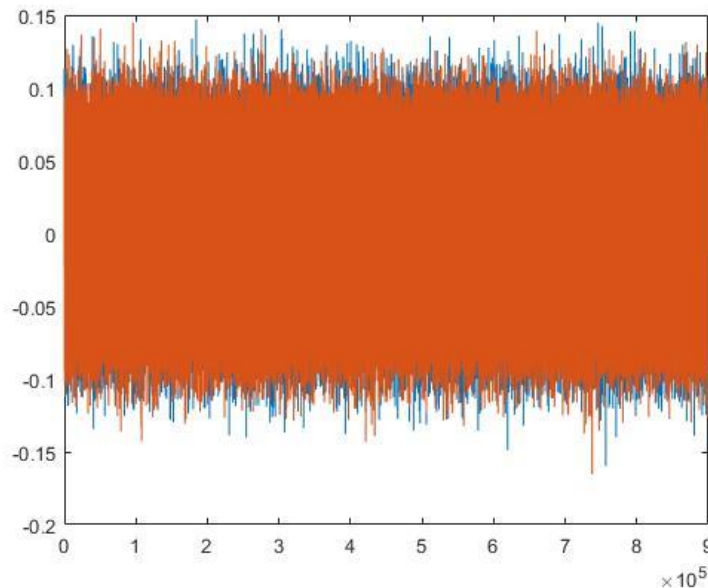
## 1.6 Συνάρτηση προσθήκης λευκού θορύβου Gaussian

Σε πρώτη φάση έπρεπε να φτιάξουμε ένα κενό πίνακα (με μηδενικά) με ίδιες διαστάσεις σαν αυτές του ηχητικού σήματος. Έπειτα να χρησιμοποιήσουμε την συνάρτηση `awgn` με παραμέτρους τον πίνακα που φτιάξαμε και την τιμή 30dB.

- `y = zeros(900000,2)`
- `y = awgn(y,30)`
- `plot(y)`



Έτσι καταφέραμε να απεικονίσουμε το θόρυβο που θα προσθέσουμε εν τέλει στο αρχικό ηχητικό μας σήμα. Με διαφορετικά χρώματα αναπαρίστανται τα 2 κανάλια.



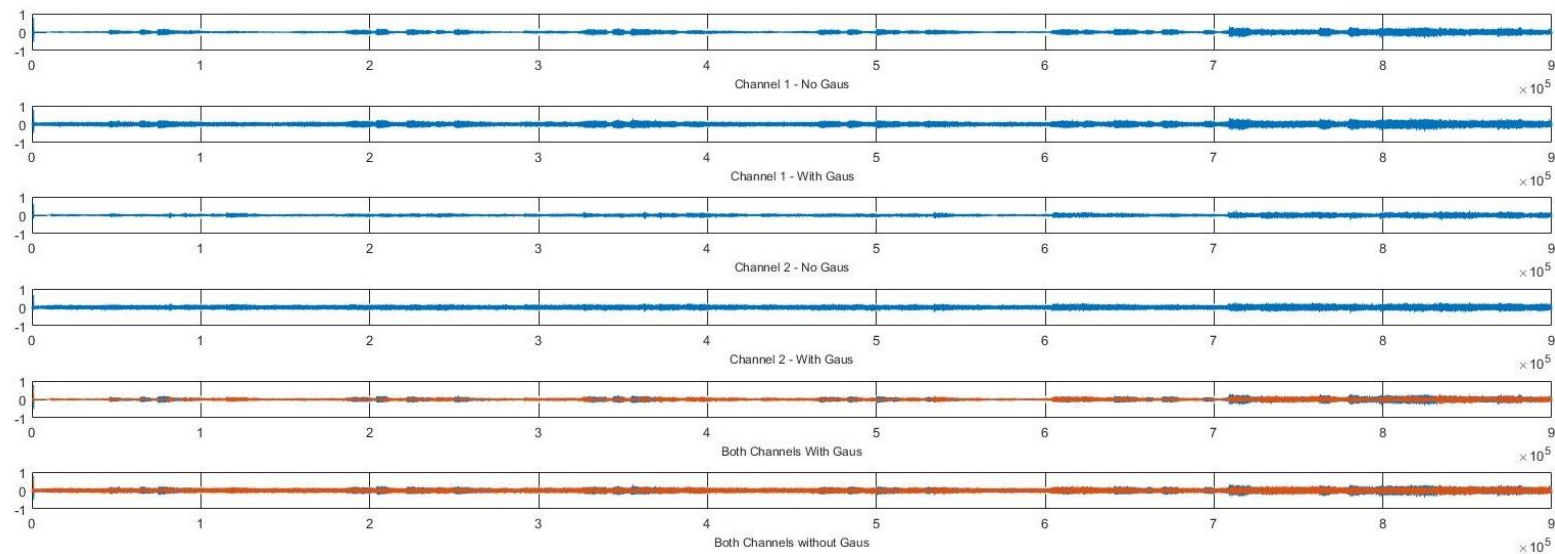
Εικόνα 1.6.1 : Γραφική αναπαράσταση λευκού θορύβου (30dB)

```
1 function [ z,y ] = gaus( x )
2
3 z=zeros(900000,2); %Δημιουργία πίνακα για
4 snr = 30;          %προσθήκη θορύβου σε αυτον|
5
6 z= awgn(z,snr); %Σήμα θορύβου
7 plot(z);
8
9 y = awgn(x,snr); %Τελικό σήμα με θόρυβο Gaus
10
11 figure; subplot(6,1,1); plot(x(:,1)); subplot(6,1,2);plot(y(:,1));subplot(6,1,3); plot(x(:,2)); ...
12 subplot(6,1,4); plot(y(:,2)); subplot(6,1,5); plot(x); subplot(6,1,6); plot(y);
13 end
```

Εικόνα 1.6.2 : Συνάρτηση πρόσθεσης λευκού θορύβου

Για το τελικό figure έχουμε 6 παραστάσεις.

- Channel 1 – No Gaus
- Channel 1 – With Gaus
- Channel 2 – No Gaus
- Channel 2 – With Gaus
- Και τα 2 κανάλια χωρίς θόρυβο Gaus
- Και τα 2 κανάλια με θόρυβο Gaus



Εικόνα 1.6.3 : Ολική γραφική παράσταση

Ως εξόδους εκτός από το τελικό σήμα που έχει προστεθεί σαντό θόρυβος φροντίσαμε να βάλουμε και άλλη μία παράμετρο στην οποία επιστρέφουμε τον θόρυβο ώστε να τον ακούσουμε. Αυτό που παρατηρούμε για τις ζητούμενες περιπτώσεις , θόρυβος στο κάθε κανάλι ξεχωριστά και ταυτόχρονα και στα 2, είναι ότι όταν προσθέτουμε θόρυβο σε ένα μόνο κανάλι τότε ο θόρυβος αναπαράγεται μόνο από την αντίστοιχη πλευρά των ακουστικών. Όταν προστίθεται όμως και στα 2 κανάλια τότε ο θόρυβος ακούγεται και στα 2.

## 1.7 Αρχείο “subject1.mat”

- gauss\_sample: περιλαμβάνει τον θόρυβο που προσθέσαμε
- x\_with\_noise: περιλαμβάνει τον αρχικό ήχο μαζί με τον προστιθέμενο θόρυβο
- faded\_in: περιλαμβάνει τον αρχικό ήχο που αυξάνεται γραμμικά
- faded\_out: περιλαμβάνει τον αρχικό ήχο που μειώνεται γραμμικά



## 2 Ζήτημα 2<sup>ο</sup>

Στο 2<sup>ο</sup> ζήτημα μας ζητείται να δημιουργήσουμε μία κενή εικόνα και να πειραματιστούμε κάνοντας υποδειγματοληψία σε αυτήν, καθώς και προσπάθεια επαναφοράς στην αρχική της κατάσταση.

### 2.1 Συνάρτηση δημιουργίας εικόνας με ομόκεντρους κύκλους

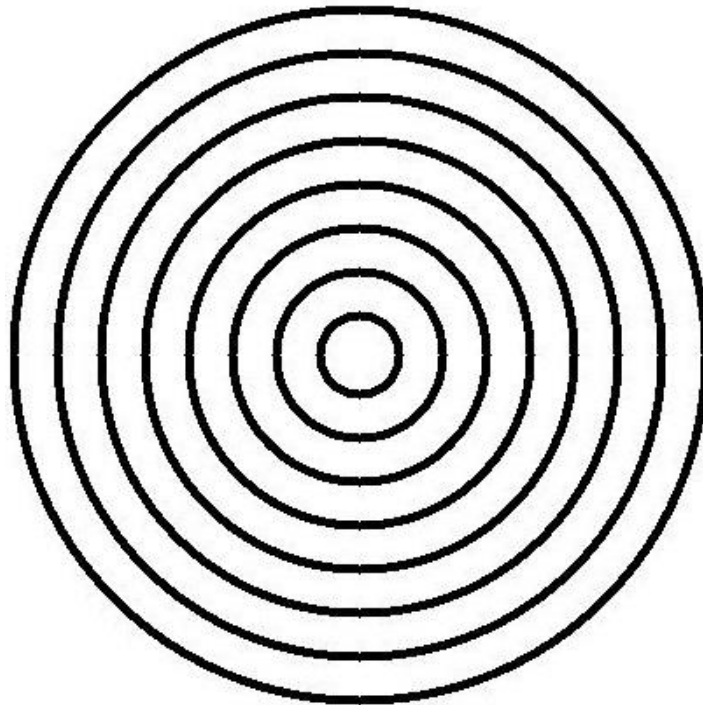
Για την συνάρτηση αυτή αρχικά φτιάξαμε έναν πίνακα στον οποίο θα θέσουμε τις αποστάσεις των pixels της εικόνας από το κέντρο της. Έπειτα υπολογίσαμε το πλήθος των κύκλων. Θα μπορούσαμε να ορίσουμε και έναν αυθαίρετο ακέραιο. Για παράδειγμα για  $N=20$  και  $M=5$  παίρνουμε 8 κύκλους στην εικόνα. Αν όμως ορίζαμε το  $a=10$  θα παίρναμε 10 κύκλους, με την διαφορά ότι οι 2 εξωτερικοί κύκλοι θα είναι ημιτελείς (Εικόνα 2.1.3). Έπειτα διατρέχουμε κάθε pixel για να υπολογίσουμε την ευκλείδεια απόσταση και να ελέγξουμε αν είναι ένα από τα pixels που πρέπει να χρωματιστεί. Αν ναι τότε χρωματίζουμε με μαύρο εκείνο το pixel. Τέλος εμφανίζουμε την εικόνα στον χρήστη. Να παρατηρήσουμε ότι για να φτιάξουμε την αρχική λευκή εικόνα κάναμε

“whitImage = 255 \* ones (400,400 ,’double’);”

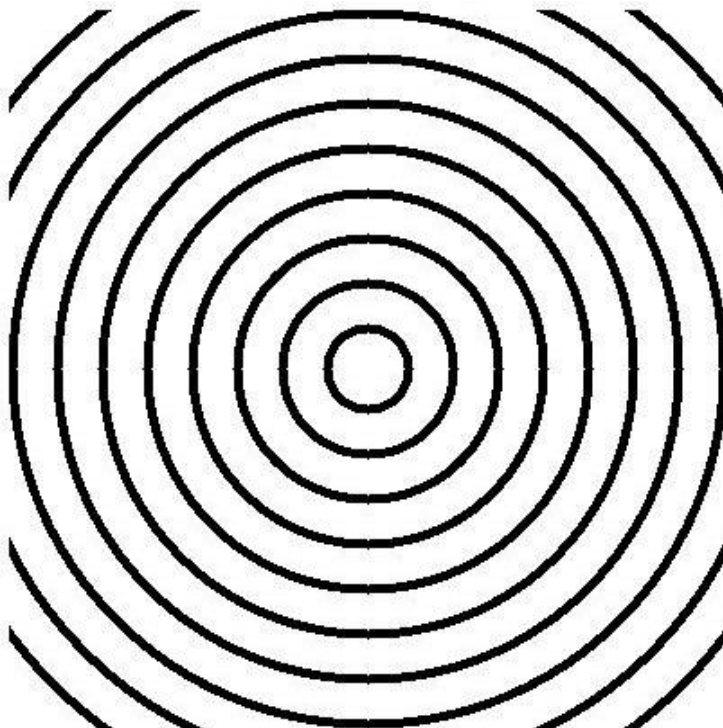
```
1 function [image] = circles( image , N , M )
2 d = zeros(400,400); %Πίνακας αποστάσεων για όλα τα pixel
3 a = 200/(M+N); %πληθος κυκλων
4
5 for i=1:a
6     for x = 1 : 400
7         for y = 1 : 400
8             z = (x-200).^2;
9             c = (y-200).^2;
10
11             d (x,y) = sqrt( z + c ); %υπολογισμος ευκλειδεις αποστασης
12
13             if d(x,y) > (i-1)*(N+M)+N && d(x,y) <= (i-1)*(N+M)+N+M %ελεγχος αποστασης pixel για βαψιμο ή μη
14                 image(x,y) = 0; %βαψε μαυρο
15                 drawnow;
16             end
17         end
18     end
19 end
20
21 imshow(image); %Εμφανιση εικονας
22 end
```

Εικόνα 2.1.1 : Συνάρτηση δημιουργίας εικόνας





Εικόνα 2.1.2 : Αρχική εικόνα για  $a=8, N=20, M=5$



Εικόνα 2.1.3 : Παραλλαγή αρχικής εικόνα για  $a=10, N=20, M=5$

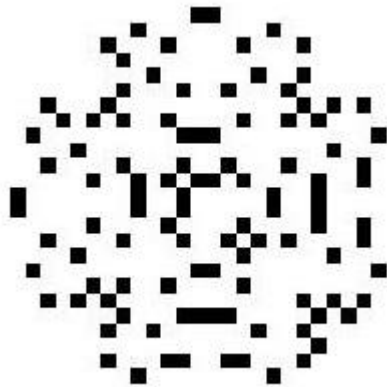


## 2.2 Υποδειγματοληψία εικόνας

Για να κάνουμε υποδειγματοληψία στην εικόνα το μόνο που κάνουμε είναι να κρατάμε κομμάτι αυτής κατά  $n$  φορές λιγότερες. Στην περίπτωση μας εμείς επιλέξαμε βήμα 16 (pixel).

```
1 function [image] = subsample( image )
2 -     newImage = image(1:16:end,1:16:end,:); %Υποδειγματολειτουργούμε κατά 16
3 -     imshow(newImage);                    %Τελικό μέγεθος εικόνας 400/16 X 400/16
4 - end
```

Εικόνα 2.2.1 : Υποδειγματοληψία κατά 16 pixel



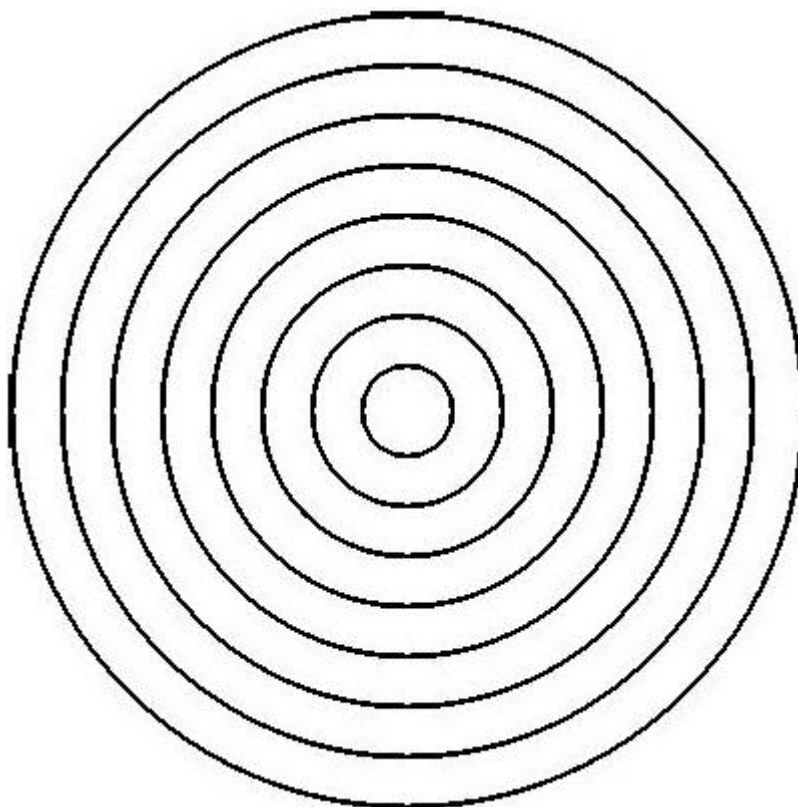
Εικόνα 2.2.2 : Αλλοιωμένη εικόνα κατά 16 φορές

## 2.3 Συνάρτηση εξάλειψης παραμόρφωσης

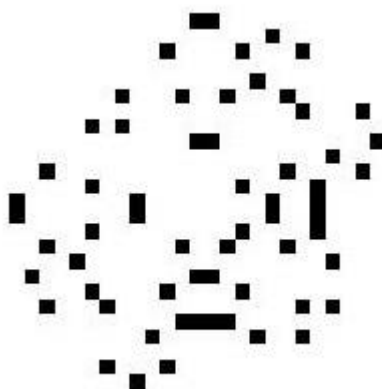
Για την συνάρτηση εξάλειψης της αλλοίωσης χρησιμοποιήσαμε όπως μας ζητήθηκε φίλτρου μέσου όρου. Για να γίνει η εφαρμογή του φίλτρου πάνω στην εικόνα πρέπει να γίνει συνέλιξη των 2 αυτών διανυσμάτων. Για τον λόγο αυτό , και επειδή τα διανύσματα μας είναι 2-D, επιλέξαμε να κάνουμε χρήση της συνάρτησης 'conv2()', με παραμέτρους την εικόνα και το φίλτρο. Στο σημείο αυτό να παρατηρήσουμε ότι ο ορισμός της εικόνας ως double κατά την δημιουργία δεν έγινε τυχαία , αλλά επειδή η συνάρτηση conv2 παίρνει ως παραμέτρους τύπους μεταβλητών 'double,double' ή 'single,single'. Στην περίπτωση που ορίζαμε την εικόνα ως 'uint8' θα έπρεπε μετέπειτα να γίνει μετατροπή αυτής σε double με την συνάρτηση 'im2double()'. [Βλ. 2.1](#)

```
1 function [ c ] = filter_avg( image )
2
3 -     a = fspecial ('average'); %φίλτρο 3x3 μεσων ορων
4 -     c = conv2(image,a,'same'); %συνέλιξη εικόνας και φίλτρου
5 -     imshow(c);
6
7 - end
```

Εικόνα 2.3.1 : Συνάρτηση δημιουργίας φίλτρου εξάλειψης παραμόρφωσης



Εικόνα 2.3.2 : Εικόνα με το πέρασμα από φίλτρο



Εικόνα 2.3.3 : Φιλτραρισμένη εικόνα μετά από υποδειγματοληψία



Όπως βλέπουμε παραπάνω έχουμε την Εικόνα 2.3.3 , η οποία είναι η αρχική εικόνα περασμένη από φίλτρο μέσου όρο (3x3) και στην συνέχεια έγινε υποδειγματοληψία πάνω της κατά 16 pixel. Επίσης έχουμε και την Εικόνα 2.2.2 , η οποία και αυτή έχει ως βάση την αρχική εικόνα. Σε αντίθεση με την προηγούμενη εικόνα υπέστη ίδια δειγματοληψία αλλά χωρίς να έχει περαστεί από το φίλτρο. Είναι εμφανείς οι διαφορές γιατί όπως παρατηρούμε έχει υποστεί και αλλοίωση στο σχήμα του περιεχομένου της.

## 2.4 Αρχείο “subject2.mat”

- N : απόσταση από τον επόμενο κύκλο (20)
- M : πάχος κύκλου (5)
- whiteImage: αρχική (λευκή) εικόνα
- image: εικόνα με τους ομόκεντρους κύκλους
- corrupt: αρχική εικόνα μετά την υποδειγματοληψία
- filtered: αρχική εικόνα μετά το φιλτράρισμα
- corrupt\_filtered: η φιλτραρισμένη εικόνα μετά από υποδειγματοληψία