

# Digital Signal Processing for Medical Imaging Using Matlab

E. S. Gopi

# Digital Signal Processing for Medical Imaging Using Matlab

E. S. Gopi  
Department of Electronics  
and Communications Engineering  
National Institute of Technology Trichy  
Tiruchirappalli, Tamil Nadu  
India

ISBN 978-1-4614-3139-8                      ISBN 978-1-4614-3140-4 (eBook)  
DOI 10.1007/978-1-4614-3140-4  
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2012944390

© Springer Science+Business Media New York 2013

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media ([www.springer.com](http://www.springer.com))

*Dedicated to my wife G. Viji, son A. G. Vasig  
and daughter A. G. Desna*

# Preface

Digital signal processing (DSP) techniques, like Radon transformation, Projection techniques, Fourier transformation in polar form, Hankel transformation, etc., are used in Medical imaging techniques like Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) during the process of imaging. These are not usually covered in the regular DSP and Image processing books. This book is written with the intention to focus the DSP aspects used during the process of imaging in CT and MRI. Also, DSP aspects used in the post imaging techniques such as Image enhancement, Image compression and pattern recognition are also discussed in this book. The Matlab illustrations are given for better understanding. This book is suitable for beginners who are doing research in Medical imaging processing.

# Acknowledgments

I am very much thankful to Prof. P. Palanisamy, Department of ECE, National Institute of Technology, Trichy, for his encouragement. I am extremely happy to express my thanks to Prof. K. M. M. Prabhu (IITM), Prof. M. Chidambaram (IITM), Prof. S. Sundararajan (NITT), Prof. P. Somaskandan (NITT), Prof. B. Venkataramani (NITT), and Prof. S. Raghavan (NITT) for their support. I also thank those who were directly or indirectly involved in bringing up this book successfully. Special thanks to my parents Mr. E. Sankara Subbu and Mrs. E. S. Meena.

# Contents

<b>1 Radon Transformation</b> . . . . .	1
1.1 Introduction to Computed Tomography (CT) . . . . .	1
1.2 Parallel Beam Projection . . . . .	1
1.2.1 Discrete Realization of (1.15). . . . .	5
1.2.2 List of Figs. 1.1 to 1.11 in Terms of the Notations Used . . . . .	6
1.3 Fanbeam Projection . . . . .	9
1.3.1 Relationship Between Parallel Beam and Fanbeam Projection. . . . .	10
1.3.2 Discrete Realization of (1.25). . . . .	15
1.3.3 List of Figs. 1.12 to 1.25 in Terms of the Notations used. . . . .	17
<b>2 Magnetic Resonance Imaging</b> . . . . .	27
2.1 Bloch Equation . . . . .	27
2.2 Comment on the Equations (2.8)–(2.10). . . . .	29
2.3 The Larmor Frequency and the Tip Angle $\alpha$ . . . . .	29
2.3.1 Disturbance to Obtain Non-Zero $\alpha$ Value. . . . .	30
2.3.2 Observation on (2.22) and (2.25) . . . . .	33
2.4 Trick on MRI . . . . .	35
2.5 Selecting the Human Slice and the Corresponding External RF Pulse . . . . .	35
2.5.1 Summary of the Section 2.5 . . . . .	38
2.6 Measurement of the Transverse Component Using the Receiver Antenna. . . . .	39
2.6.1 Observation on (2.47)–(2.50) . . . . .	40
2.6.2 Receiver to Receive the Transverse Component . . . . .	40
2.7 Sampling the MRI Image in the Frequency Domain . . . . .	41

2.8	Practical Difficulties and Remedies in MRI . . . . .	42
2.8.1	Proton-Density MRI Image Using Gradient Echo . . . . .	43
2.8.2	$T_2$ MRI Image Using Spin-Echo and Cartesian Scanning. . . . .	44
2.8.3	$T_2$ MRI Image Using Spin-Echo and Polar Scanning . . . . .	46
2.8.4	$T_1$ MRI Image . . . . .	47
<b>3</b>	<b>Illustrations on MRI Techniques Using Matlab . . . . .</b>	<b>49</b>
3.1	Illustration on the Steps Involved in Obtaining Proton-Density MRI Image . . . . .	49
3.1.1	Proton-Density MRI Imaging . . . . .	50
3.2	Illustration on the Steps Involved in Obtaining the $T_2$ MRI Image Using Cartesian Scanning . . . . .	53
3.2.1	Note to the Fig. 3.4. . . . .	57
3.2.2	Momentary Peak Due to Spin Echo . . . . .	60
3.3	Illustration on the Steps Involved in Obtaining the $T_2$ MRI Image Using Polar Scanning . . . . .	63
3.3.1	Reconstructing $f(x,y)$ from $G(r,\theta)$ . . . . .	64
3.4	Illustration on the Steps Involved in Obtaining the $T_1$ MRI Image . . . . .	68
3.4.1	t1.m . . . . .	70
<b>4</b>	<b>Medical Image Processing . . . . .</b>	<b>73</b>
4.1	Summary on the Various Medical Imaging Techniques . . . . .	73
4.2	Image Enhancement. . . . .	74
4.2.1	Logarithmic Display . . . . .	74
4.2.2	Non-Linear Filtering . . . . .	74
4.2.3	Image Substraction . . . . .	74
4.2.4	Linear Filterereng and the Hankel Transformation. . . . .	76
4.2.5	Histogram Equalization . . . . .	80
4.2.6	Histogram Specification. . . . .	81
4.3	Image Compression . . . . .	82
4.3.1	Discrete Cosine Transformation (DCT) . . . . .	82
4.3.2	Using KL-Transformation . . . . .	85
4.4	Feature Extraction and Classification . . . . .	86
4.4.1	Using Discrete Wavelet Transformation. . . . .	87
4.4.2	Dimensionality Reduction Using Principal Component Analysis (PCA). . . . .	89
4.4.3	Dimensionality Reduction Using Linear Discriminant Analysis (LDA) . . . . .	93
4.4.4	Dimensionality Reduction Using Kernel-Linear Discriminant Analysis (K-LDA) . . . . .	96



Contents	xiii
<b>Appendix A: Solving Bloch Equation with <math>A\Delta v \text{ sinc}(\Delta vt)</math> Envelope . . .</b>	<b>101</b>
<b>Appendix B: Projection Techniques . . . . .</b>	<b>103</b>
<b>Appendix C: Hankel Transformation . . . . .</b>	<b>107</b>
<b>Appendix D: List of m-Files . . . . .</b>	<b>109</b>
<b>Index . . . . .</b>	<b>111</b>

# Chapter 1

## Radon Transformation

### 1.1 Introduction to Computed Tomography (CT)

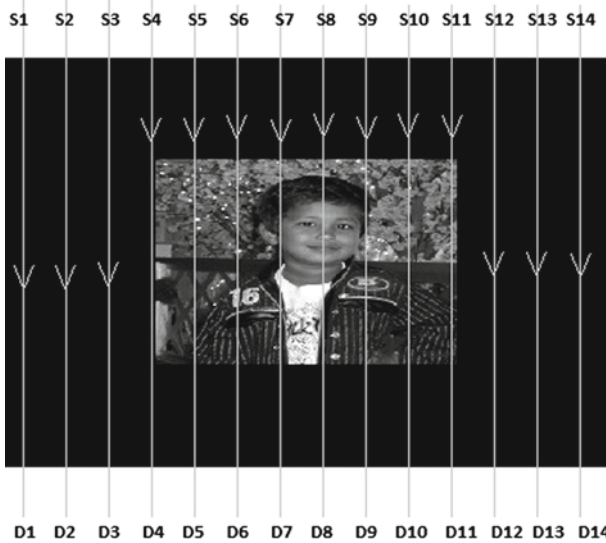
The physical setup to obtain the CT of the particular slice of the test body involves passing the X-ray to that particular slice and detecting the attenuated signal at the other side. This value is conceptually proportional to the integral value of sliced image along the X-ray paths. The ray-path directions with respect to sliced image describes the type of projection used in that CT. There are two major types of projection techniques namely Parallel beam projection and Fan-beam projection used in CT. The process of reconstructing the image from the projected data involves digital signal processing, which are described below.

### 1.2 Parallel Beam Projection

Let us consider an example image (refer Fig. 1.1) as the sliced image of the test body. The parallel beam projection involves transmitting X-ray signals one after and another, parallel to each other and the corresponding attenuated signals are captured using the detector kept exactly on the other sides of the ray (refer Fig. 1.2). This is equivalent to obtaining the line integration of the image in the direction of the parallel beam. This is the radon transformation with an angle  $0^\circ$ . Now the image is rotated in the clock-wise direction with an angle  $\theta^\circ$  and the line integration is computed as mentioned above. This is the radon transformation with angle  $\theta^\circ$ . (In practice, this is obtained by shifting the positions of the source and the detector such that the imaginary line joining the source and the detector is rotated in the anticlockwise direction by an angle  $\theta^\circ$ ). This is repeated for the angle  $\theta^\circ$  ranging from 0 to  $360^\circ$ . This completes the forward radon transformation. The process of estimating the original image from the forward radon transformation data is called as inverse radon transformation, which is described below.



**Fig. 1.1** Original image (A) of size  $511 \times 511$  subjected to parallel and fan-beam radon transformation



**Fig. 1.2** Parallel beam projection geometry

Let  $f(x, y)$  be the image in the rectangular co-ordinate system, where  $x, y$  ranges from  $-\infty$  to  $\infty$ . Let the centre of the image be at the position  $(0, 0)$ . Let the set of parallel lines tilted with an angle  $\theta$  anticlockwise direction is represented as  $x \cos \theta + y \sin \theta = l$ . Note that for different values of  $l$ , we get different lines that are parallel to each other. so radon transformation with an angle  $\theta$  can be represented as follows.

$$R(l, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - l) dx dy \quad (1.1)$$

For the fixed  $\theta$ ,  $R(l, \theta) = R(l)$ . Taking fourier transformation of  $R(l)$ , we get the following.

$$G(U) = \int_{-\infty}^{\infty} R(l) \exp^{-j2\pi U l} dl \quad (1.2)$$

$$\Rightarrow G(U) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - l) dx dy \exp^{-j2\pi U l} dl \quad (1.3)$$

$$\Rightarrow G(U) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp^{-j2\pi(x \cos \theta + y \sin \theta)U} dx dy \quad (1.4)$$

Let the 2D-Fourier transformation of the image matrix  $f(x, y)$  be represented as  $F(U', V')$ . It is noted from the (1.4) that  $G(U) = F(U', V')$ , when  $U' = U \cos \theta$  and  $V' = U \sin \theta$ . If the values are collected from the 2D-Fourier transformation  $F(U', V')$  of  $f(x, y)$ , along the line  $U' \cos(\theta) + V' \sin(\theta) = U$  (i.e) for various  $U$ , we get  $G(U)$ . This is called projection-slice theorem.

$f(x, y)$  is obtained from the  $F(U', V')$  using inverse 2D-Fourier transformation as mentioned below.

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(U', V') \exp^{j2\pi(xU' + yV')} dU' dV' \quad (1.5)$$

Substituting  $U' = U \cos \theta$  and  $V' = U \sin \theta$  in (1.5), we get the following.  $F(U \cos \theta, U \sin \theta) = G(U, \theta)$ , which is the Fourier transformation of  $g(l, \theta)$  for the constant  $\theta$  (refer (1.2)). Changing the variables from  $(x, y)$  to  $(U, \theta)$ , (1.5) becomes

$$f(x, y) = \int_{-\pi}^{\pi} \int_0^{\infty} G(U, \theta) \exp^{j2\pi(x \cos \theta + y \sin \theta)U} |J| dU d\theta \quad (1.6)$$

where  $|J|$ , is the jacobian of the transformation  $U = \sqrt{U'^2 + V'^2}$ ,  $\theta = \tan^{-1} \frac{V'}{U'}$ .

$$(i.e) J = \begin{bmatrix} \frac{\partial U}{\partial U'} & \frac{\partial U}{\partial V'} \\ \frac{\partial \theta}{\partial U'} & \frac{\partial \theta}{\partial V'} \end{bmatrix} \Rightarrow |J| = |U|$$

Hence,(1.6) is rewritten as

$$f(x, y) = \int_{-\pi}^{\pi} \int_0^{\infty} G(U, \theta) \exp^{j2\pi(x \cos \theta + y \sin \theta)U} |U| dU d\theta \quad (1.7)$$

From (1.1), we get  $R(l, \theta) = R(-l, \theta + \pi)$ . This implies

$$G(-U, \theta) = G(U, \theta + \pi) \quad (1.8)$$

Splitting (1.7) into two terms,

I-term:

$$\int_{-\pi}^0 \int_{-\infty}^{\infty} G(U, \theta) \exp^{j2\pi(x \cos \theta + y \sin \theta)U} |U| dU d\theta \quad (1.9)$$

II-term:

$$\int_0^{\pi} \int_0^{\infty} G(U, \theta) \exp^{j2\pi(x \cos \theta + y \sin \theta)U} |U| dU d\theta \quad (1.10)$$

Change the variable  $\phi = \theta + \pi$  and  $U1 = -U$  in the first term (1.9), we get

$$\int_0^{\pi} \int_0^{\infty} G(-U1, \phi - \pi) \exp^{j2\pi(x \cos \phi + y \sin \phi)U1} |U1| dU1 d\phi \quad (1.11)$$

Using (1.7)–(1.11), we get

$$f(x, y) = 2 \int_0^{\pi} \int_0^{\infty} G(U1, \phi - \pi + \pi) \exp^{j2\pi(x \cos \phi + y \sin \phi)U1} |U1| dU1 d\phi. \quad (1.12)$$

Replacing the dummy variables  $U1$  and  $\phi$  with  $U$  and  $\theta$  respectively in (1.12) and writing the second limit ranging from  $-\infty$  to  $\infty$ , we get the following.

$$f(x, y) = 2(1/2) \int_0^{\pi} \int_{-\infty}^{\infty} G(U, \theta) \exp^{j2\pi(x \cos \theta + y \sin \theta)U} |U| dU d\theta \quad (1.13)$$

Let  $l = x \cos \theta + y \sin \theta$ . Thus rewriting (1.12) as follows.

$$f(x, y) = \int_0^{\pi} \int_{-\infty}^{\infty} G(U, \theta) \exp^{j2\pi l U} |U| dU d\theta \quad (1.14)$$

Note that  $\int_{-\infty}^{\infty} G(U, \theta) \exp^{j2\pi lU} |U| dU$  is the inverse fourier transformation of the function  $G(U, \theta)|U|$  for constant  $\theta$ . This can be achieved as the convolution of inverse fourier transform (IFT) of  $G(U, \theta)$  and IFT of  $|U|$ . Note that IFT of  $G(U, \theta)$  is  $R(l, \theta)$  for constant  $\theta$  (refer (1.2)). Final reconstruction formula from parallel beam radon transformation  $R(l, \theta)$  is represented as follows.

$$f(x, y) = \int_0^{\pi} R(l, \theta) * IFT(|U|) d\theta \quad (1.15)$$

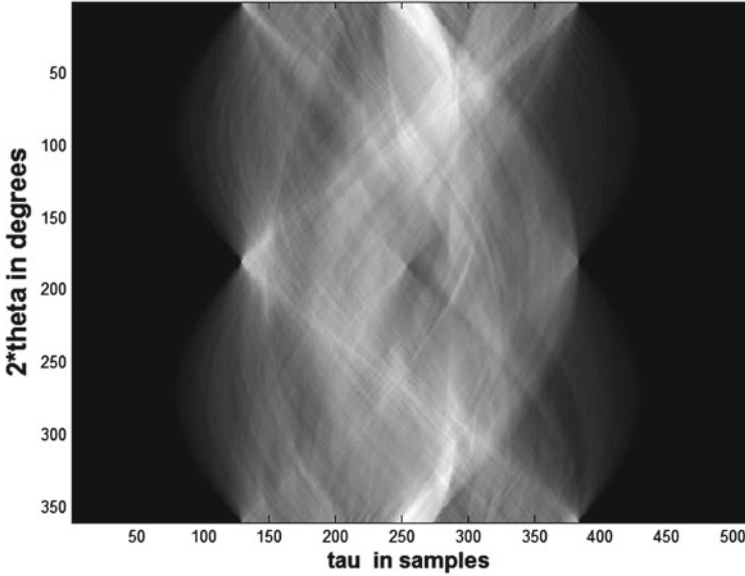
### 1.2.1 Discrete Realization of (1.15)

Let  $Z(\theta, x, y)$  be the value obtained by the convolution  $R(l, \theta) * IFT(|U|)$  at  $l = x \cos \theta + y \sin \theta$  for the particular  $\theta$ ,  $x$  and  $y$ . Integrating  $Z(\theta, x, y)$  over the complete range of  $\theta$  (0 to  $\pi$ ) with constant  $x$  and  $y$  gives the value of  $f$  at  $(x, y)$ . Let the discrete version of the continuous image  $f$  be represented as  $f_d$ . Note that  $f_d$  is the image matrix that are having finite range for  $x$  and  $y$  with origin in the middle of the image. It is noted that the set of co-ordinates  $(x, y)$  whose  $l$  is constant for the particular  $\theta$  is the straight line  $x \cos(\theta) + y \sin(\theta) = l$ . Thus to obtain  $f_d$ , the following steps are followed.

1. Create the zero matrix  $r$  whose size is exactly same as that of  $f_d$ .
2. Let  $z(l, \theta)$  be the value obtained by the convolution  $R(l, \theta) * IFT(|U|)$  at  $l$  for the particular  $\theta$ .
3. Fill the matrix  $r$  with  $z(l, \theta)$  in all the co-ordinates  $(x, y)$  that satisfies  $x \cos(\theta) + y \sin(\theta) = l$ , which is the straight line tilted with an angle  $\theta$  in the anticlockwise direction.
4. Repeat step 2 for all  $l$  ranging from  $-l_{\max}$  to  $l_{\max}$  with fixed  $\theta$ . Let the obtained matrix be represented as  $r_{\theta}$ . Note that the lines corresponding to different  $l$  for the particular  $\theta$  are parallel to each other.
5. Compute  $r_{\theta}$  for all values of  $\theta$  with some resolution for  $\theta$ .
6. Thus  $f_d$  (discrete version of  $f$ ) is obtained as  $\sum_{\theta=0}^{\theta=\pi} r_{\theta}$ .

Instead of filling the matrix with the particular value in the co-ordinates corresponding to the particular line tilted with an angle  $\theta$  anticlockwise, fill the matrix in the particular column and tilt the matrix by an angle  $\theta$  anticlockwise. Thus the matrix  $f_d$  can also be obtained as follows.

1. Create the zero matrix  $r$ .
2. Fill the first row of the matrix  $r$  with the vector  $R(l, \theta) * IFT(|U|)$ .
3. All other rows of the matrix  $r$  are also filled up with the same vectors.
4. Rotate the matrix  $r$  by an angle  $\theta$  anticlockwise to obtain  $r_{\theta}$ .
5. Compute  $r_{\theta}$  for all values of  $\theta$  with some resolution for  $\theta$ .
6. Thus  $f(x, y)$  is obtained as  $\sum_{\theta=0}^{\theta=\pi} r_{\theta}$ .



**Fig. 1.3** Parallel beam radon transformation of the image ‘A’ (refer Fig.1.1) with theta resolution =  $0.5014^\circ$

### 1.2.2 List of Figs. 1.1 to 1.11 in Terms of the Notations Used

- Figure 1.1: Original image subjected to parallel beam projection.
- Figure 1.2: Parallel beam projection geometry
- Figure 1.3:  $R(l, \theta)$  (Sinogram) corresponding to the original image with the resolution of  $\theta = 24.5682$ .
- Figure 1.4: Impulse response and the transfer function of the filter  $|U|$
- Figure 1.5:  $R(l, \theta)$  for  $\theta = 24.5682$  degree and  $R(l, \theta) * |U|$
- Figure 1.6:  $r_\theta$  for various  $\theta$  with  $R(l, \theta) * IFT(|U|) = R(l, \theta)$  ((i.e.) without filtering)
- Figure 1.7:  $r_\theta$  for various  $\theta$  (with filtering)
- Figure 1.8:  $\sum_{\theta=0}^{\theta=\Theta} r_\theta$  for various  $\Theta$  with  $R(l, \theta) * IFT(|U|) = R(l, \theta)$  ((i.e.) without filtering)
- Figure 1.9:  $\sum_{\theta=0}^{\theta=\Theta} r_\theta$  for various  $\Theta$  ((i.e.)without filtering)
- Figure 1.10: Final reconstructed image  $f_d$  obtained with  $R(l, \theta) * IFT(|U|) = R(l, \theta)$  ((i.e.) without filtering)
- Figure 1.11: Final reconstructed image  $f_d$  (with filtering)

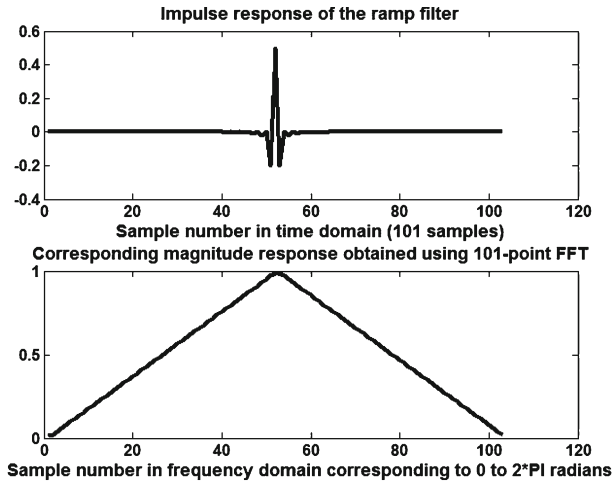


Fig. 1.4 Impulse response of the ramp filter and the corresponding spectrum

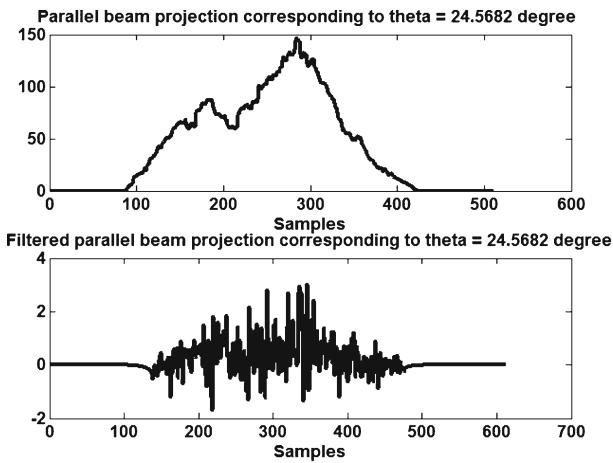


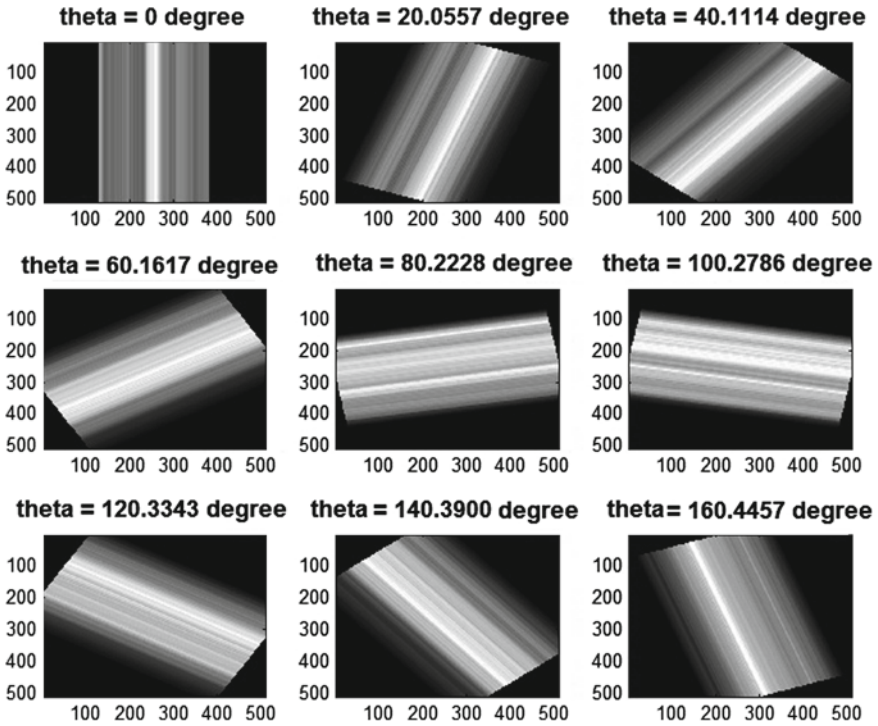
Fig. 1.5 Original and the corresponding filtered parallel beam projection for  $\theta = 24.5682^\circ$

1.2.2.1 Parallelbeamprojection.m

```

RECONSTRUCTEDIMAGE=0;
load VASIGIMAGE
C=C(1:1:255,1:1:255);
C=[zeros(128,511);zeros(255,128) C zeros(255,128);zeros
(128,511)];
B=double(C)/255;
T=1;
    
```



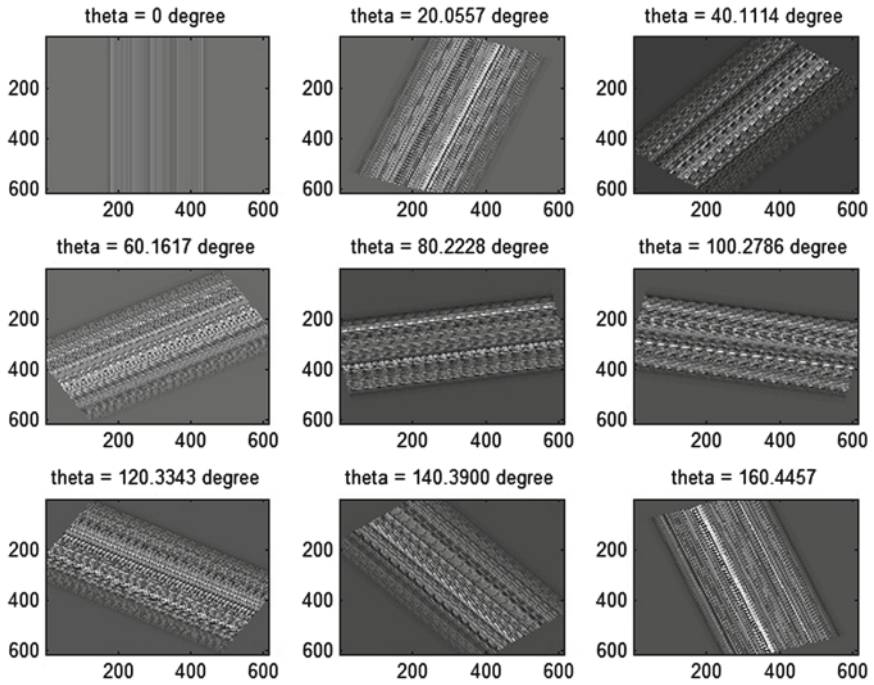


**Fig. 1.6** Illustration of original (without filtering) parallel beam backprojected images obtained for  $\theta = 0, 20.0557, 40.1114, 60.1617, 80.2228, 100.2786, 120.3343, 140.3900, 160.4457^\circ$  (Note that parallel-beam reconstructed images are computed for all  $\theta$  ranging from  $0$  to  $180^\circ$  with the resolution of  $0.5014^\circ$ )

```

angleresolution=360;
for theta=0:(180/(angleresolution-1)):180
%disp( theta)
RADONTRANSFORMATION{T}=sum(imrotate(B,theta,'nearest','crop'));
T=T+1;
end
%Back projection technique to get back the data
S=1;
Z=[1:10:359];
Z=[Z 0];
z=1;
k=0;
for theta=0:(180/(angleresolution-1)):180
k=k+1;
disp( theta)
%Filtering with ramp spectrum
temp=conv(RADONTRANSFORMATION{S},fir2(102,(0:1/101:1),
(0:1/101:1),rectwin(103)));
temp=repmat(temp,size(temp,2),1)';

```

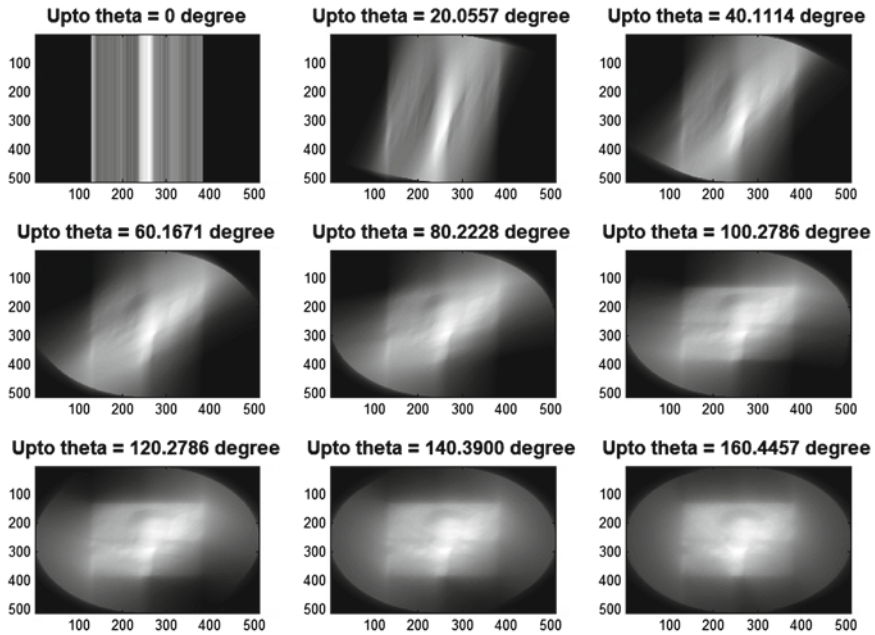


**Fig. 1.7** Illustration of filtered parallel beam backprojected images obtained for  $\theta = 0, 20.0557, 40.1114, 60.1617, 80.2228, 100.2786, 120.3343, 140.3900, 160.4457^\circ$  (Note that parallel-beam reconstructed images are computed for all  $\theta$  ranging from 0 to  $180^\circ$  with the resolution of  $0.5014^\circ$ )

```

temp=imrotate(temp,90);
D=imrotate(temp,-theta,'nearest','crop');
RECONSTRUCTEDIMAGE=RECONSTRUCTEDIMAGE+D;
if (Z(z)==k)
    SNAPSHOT_RECONSTRUCTED{z}=RECONSTRUCTEDIMAGE;
    SNAPSHOT_DATA{z}=D;
    z=z+1;
end
S=S+1;
end
figure
colormap(gray(256))
imagesc(RECONSTRUCTEDIMAGE)

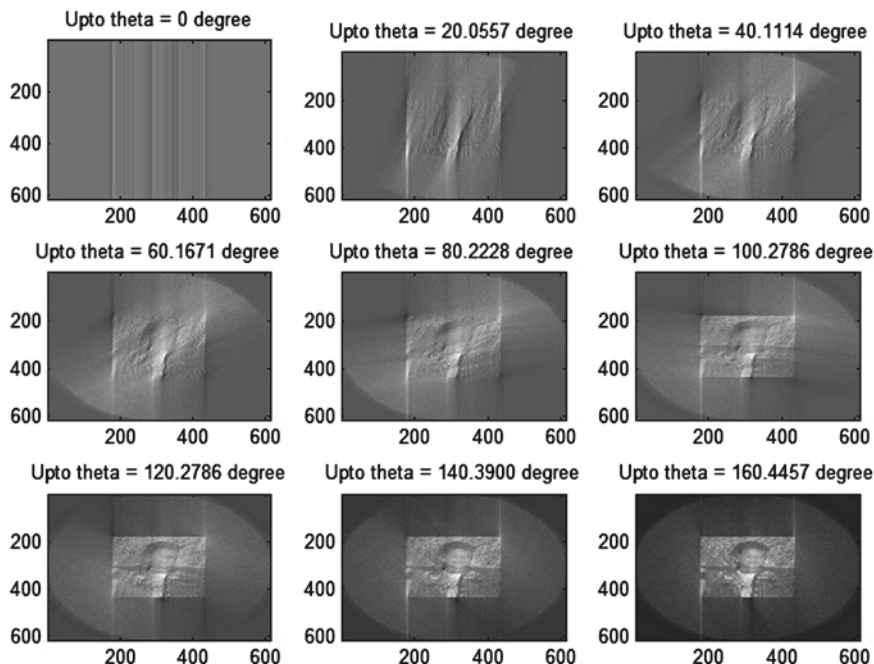
```



**Fig. 1.8** Illustration of reconstruction of the image obtained by cumulative summation of original parallel beam backprojected images

### 1.3 Fanbeam Projection

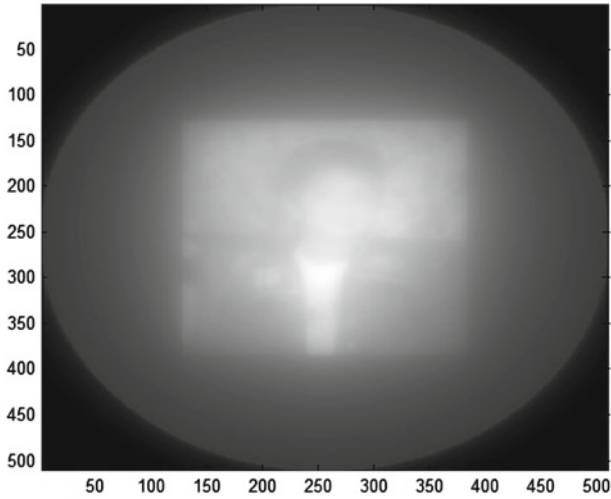
The parallel beam projection technique based CT Scan takes more scanning time. Hence Fanbeam projection technique based CT Scan is used. The Geometry for the Fan beam projection is as mentioned in the Fig. 1.12. The single X-ray source ( $S$ ) kept in the  $y$ -axis at the distance  $D$  from the origin is subjected to the slice of the test body and the multiple detectors ( $D1-D5$ ) kept at the circumference of the sector (obtained with  $S$  as the centre) are used to capture the attenuated signal along the ray of paths. Mathematically this is proportional to the line integrals taken along the ray of paths. This arrangement helps in reducing the scanning time. This corresponds to the angle  $\beta = 0$ . Further the source is tilted by an angle  $\beta = B$  in the anticlockwise direction and the corresponding attenuated signals are captured. This is repeated for  $\beta$  ranging from  $0$  to  $2\pi$  radians. Positions of the detectors are described by the angle measured from reference line segment  $SD$  to the line segment joining the source( $S$ ) and the detector in the anticlockwise direction. This angle is represented as  $\delta$ . Detector 1 (refer Fig. 1.12), is kept at an angle  $-\delta_{\max}$  and the detector 5 is kept at an angle  $+\delta_{\max}$ .



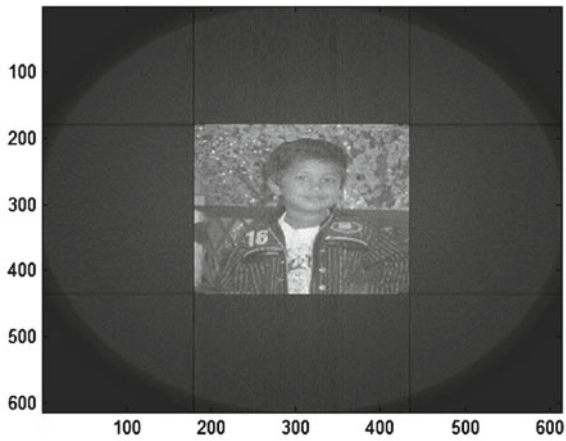
**Fig. 1.9** Illustration of reconstruction of the image obtained by cumulative summation of filtered parallel beam backprojected images

### ***1.3.1 Relationship Between Parallel Beam and Fanbeam Projection***

Conceptually (refer Sect. 1.2 and Fig. 1.13), the  $\theta$  used in parallel beam projection is the angle (measured in the anticlockwise direction) between the Y-axis and line segment joining the source and the detector ( $SD$ ). Also  $l$  is the perpendicular distance with sign measured from the origin to the line segment  $SD$ . The line integration obtained from the line segment  $SD$  in Fig. 1.14 corresponds to  $\beta = 0$  and  $\delta = -\delta_1$  while using Fan beam projection. The same line integration, if it is computed using the parallel beam projection, corresponds to  $-\theta = \delta_1$  and  $l = C \sin \delta_1$ . Note that the relationship holds for  $\beta = 0$ . If the reference line segment is rotated by a non-zero angle  $\beta$  in the anticlockwise direction (refer Fig. 1.14), we get  $l = C \sin(-\delta_1)$  and  $\theta = B - \delta_1$ . In general, line integration obtained from the fanbeam projection with the particular  $\delta$  and  $\beta$  equals to the line integration obtained from the parallel beam projection with  $\theta = \beta + \delta$  and  $l = C \sin(\delta)$ . Thus the reconstruction formula used for parallel beam projection technique can be used for formulating the reconstruction formula for Fan beam projection technique as described below.



**Fig. 1.10** Final reconstructed image obtained from parallel beam without filtering

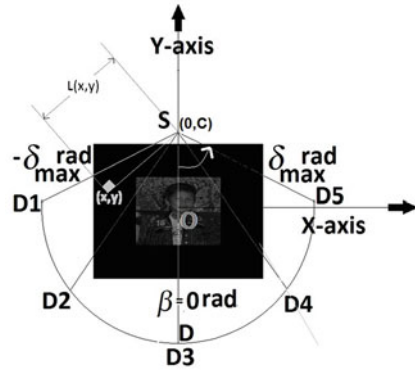


**Fig. 1.11** Final reconstructed image obtained from filtered parallel beam

Rewriting (1.15) with  $\theta$  ranging from 0 to  $2\pi$ , we get the following.

$$f(x, y) = \frac{1}{2} \int_0^{2\pi} R(l, \theta) * IFT(|U|) d\theta \quad (1.16)$$

**Fig. 1.12** Fan-beam projection geometry



Let  $IFT(|U|)$  be represented as  $u(l)$ .

$$f(x, y) = \frac{1}{2} \int_0^{2\pi} \int_{-\infty}^{\infty} R(\tau, \theta) u(l - \tau) d\tau d\theta \tag{1.17}$$

Put  $l = x \cos(\theta) + y \sin(\theta)$ ,

$$f(x, y) = \frac{1}{2} \int_0^{2\pi} \int_{-\infty}^{\infty} R(\tau, \theta) u(x \cos(\theta) + y \sin(\theta) - \tau) d\tau d\theta \tag{1.18}$$

Replacing the dummy variable  $\tau$  with  $l$ ,

$$f(x, y) = \frac{1}{2} \int_0^{2\pi} \int_{-\infty}^{\infty} R(l, \theta) u(x \cos(\theta) + y \sin(\theta) - l) dl d\theta \tag{1.19}$$

Let the co-ordinates be represented in the polar form (i.e)  $x = r \cos(\phi)$  and  $y = r \sin(\phi)$ . Hence (1.19) becomes

$$f(r, \phi) = \frac{1}{2} \int_0^{2\pi} \int_{-\infty}^{\infty} R(l, \theta) u(r \cos(\phi) \cos(\theta) + r \sin(\phi) \sin(\theta) - l) dl d\theta \tag{1.20}$$

$$\Rightarrow f(r, \phi) = \frac{1}{2} \int_0^{2\pi} \int_{-\infty}^{\infty} R(l, \theta) u(r \cos(\phi - \theta) - l) dl d\theta \tag{1.21}$$

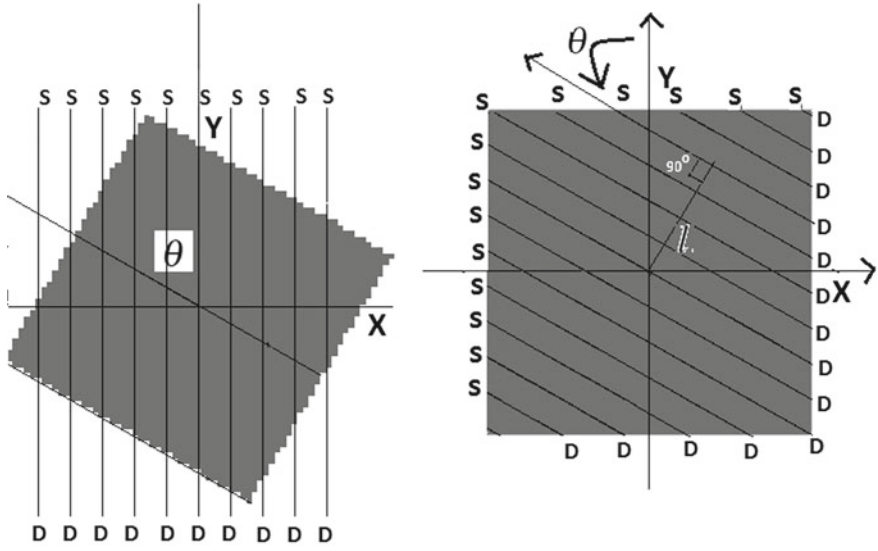


Fig. 1.13 Relationship between parallel beam and the fan-beam projection geometry-1

(1.21) is rewritten using the new set of variables  $\theta = \beta + \delta$  and  $l = C \sin(\delta)$ .

$$\begin{aligned}
 & f(r, \phi) \\
 &= \int_{-\delta_{min}}^{\delta_{max}} \int_{0+\delta}^{2\pi+\delta} R(C \sin(\delta), \beta - \delta) u(r \cos(\phi - \beta - \delta) - C \sin(\delta)) |J| d\beta d\delta,
 \end{aligned}
 \tag{1.22}$$

where  $|J|$ , is the jacobian of the transformation  $\theta = \beta + \delta$  and  $l = C \sin(\delta)$  (i.e)

$$J = \begin{bmatrix} \frac{\partial \beta}{\partial \theta} & \frac{\partial \beta}{\partial \delta} \\ \frac{\partial l}{\partial \theta} & \frac{\partial l}{\partial \delta} \end{bmatrix} \Rightarrow |J| = |C \cos(\delta)|$$

Hence, (1.22) is rewritten as

$$\begin{aligned}
 & f(r, \phi) \\
 &= \int_{-\delta_{max}}^{\delta_{max}} \int_{-\delta+0}^{-\delta+2\pi} R(C \sin(\delta), \beta + \delta) u(r \cos(\phi - \beta - \delta) - C \sin(\delta)) |C \cos(\delta)| d\beta d\delta \tag{1.23}
 \end{aligned}$$

With constant  $\delta$ ,  $R(C \sin(\delta), \beta + \delta)$  and  $u(r \cos(\phi - \beta - \delta) - C \sin(\delta))$  are the periodic functions of the variable  $\beta$  with period  $2\pi$ . Thus the final reconstruction

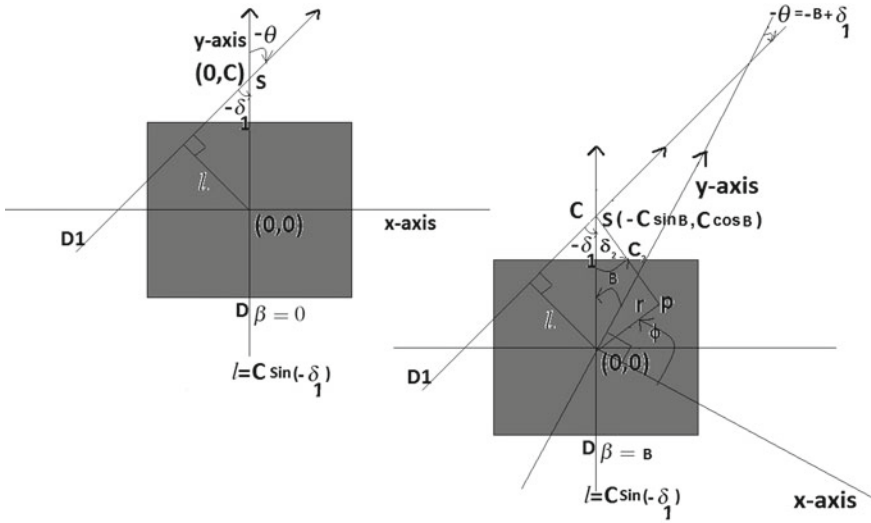


Fig. 1.14 Relationship between parallel beam and the fan-beam projection geometry-2

formula for Fanbeam projection is given below.

$$f(r, \phi) = \int_{-\delta_{\max}}^{\delta_{\max}} \int_0^{2\pi} R(C \sin(\delta), \beta + \delta) u(r \cos(\phi - \beta - \delta) - C \sin(\delta)) |C \cos(\delta)| d\beta d\delta \tag{1.24}$$

interchanging the order of the limits, we get

$$f(r, \phi) = \int_0^{2\pi} \int_{-\delta_{\max}}^{\delta_{\max}} R(C \sin(\delta), \beta + \delta) u(r \cos(\phi - \beta - \delta) - C \sin(\delta)) |C \cos(\delta)| d\delta d\beta \tag{1.25}$$

### 1.3.2 Discrete Realization of (1.25)

Consider the fan-beam ray path joining the source point  $S$  and the arbitrary point  $P$  with polar co-ordinates  $(r, \phi)$ . Let the distance between the  $S$  and  $P$  be  $C_2$  (refer Fig. 1.15). Note that the ray path corresponds to the angle  $\beta = B$  and  $\delta = \delta_2$ . From the geometry structure (refer Fig. 1.15), it is found that  $C = C_2 \cos(\delta_2) + r \cos(\phi - B)$  and  $r \cos(\phi - B) = C_2 \sin(\delta_2)$ . This is true for all values of  $\beta$ . So for some arbitrary  $\beta$ ,  $C = C_2 \cos(\delta_2) + r \cos(\phi - \beta)$  and  $r \cos(\phi - \beta) = C_2 \sin(\delta_2)$ . Substituting back in (1.25), we get the simplified version of  $r \cos(\phi - \beta - \delta) - C \sin(\delta)$  (part of 1.26)





transformation of ramp function  $|U|$ . From inverse fourier transformation we can represent

$$u(C_2 \sin(\delta_2)) = \int_{-\infty}^{\infty} |U| e^{j2\pi U C_2 \sin(\delta_2)} dU \quad (1.28)$$

Let  $U_1 \delta_2 = C_2 \sin(\delta_2) U$  substituting back in the Eq. (1.28), we get the following.

$$\int_{-\infty}^{\infty} \left| \frac{U_1 \delta_2}{C_2 \sin(\delta_2)} \right| e^{j2\pi U_1 \delta_2} \frac{\delta_2}{C_2 \sin(\delta_2)} dU_1 \quad (1.29)$$

$$\Rightarrow u(C_2 \sin(\delta_2)) = \left( \frac{\delta_2}{C_2 \sin(\delta_2)} \right)^2 \int_{-\infty}^{\infty} |U_1| e^{j2\pi U_1 \delta_2} dU_1 \quad (1.30)$$

$$\Rightarrow u(C_2 \sin(\delta_2)) = \left( \frac{\delta_2}{C_2 \sin(\delta_2)} \right)^2 u(\delta_2) \quad (1.31)$$

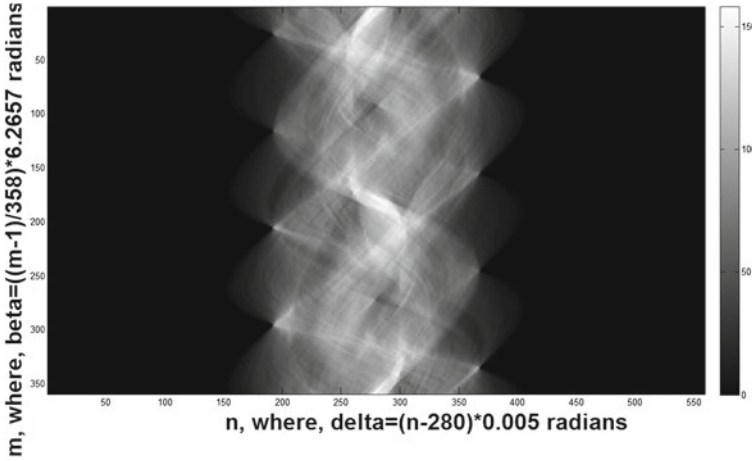
Also note that  $C_2$  is constant for the particular value of  $r$  and  $\phi$ . Rewriting (1.27) using (1.31), we get the following.

$$f(r, \phi) = \int_0^{2\pi} \int_{-\delta_{\max}}^{\delta_{\max}} W(\delta, \beta) \left( \frac{\delta_2 - \delta}{C_2 \sin(\delta_2 - \delta)} \right)^2 u(\delta_2) |C \cos(\delta)| d\delta d\beta \quad (1.32)$$

$$\Rightarrow f(r, \phi) = \int_0^{2\pi} \int_{-\delta_{\max}}^{\delta_{\max}} \frac{1}{(C_2)^2} W(\delta, \beta) \left( \frac{(\delta_2 - \delta)}{\sin(\delta_2 - \delta)} \right)^2 u(\delta_2 - \delta) |C \cos(\delta)| d\delta d\beta \quad (1.33)$$

Thus discrete form of (1.33) is realized as follows.

1. Compute the product of fan-beam radon transformation  $W(\delta, \beta)$  for the particular  $\beta$  with  $C \cos(\delta)$ . Note that  $C$  is the constant. Treat the result as the function of  $\delta_2$  (say  $fanradon(\delta_2)$ ).
2. Compute the convolution of  $fanradon(\delta_2)$  with  $\left( \frac{\delta_2}{\sin(\delta_2)} \right)^2 u(\delta_2)$  to obtain  $faniradon(\delta_2)$ . Note that the sample of  $fanradon(\delta_2)$  corresponds to the  $\delta_2$  ranging from  $-\delta_{\max}$  to  $\delta_{\max}$ . Also the samples of  $\left( \frac{\delta_2}{\sin(\delta_2)} \right)^2 u(\delta_2)$  corresponds to the  $\delta_2$  ranging from  $-\delta_{\max}$  to  $\delta_{\max}$ . Hence the index of the convoluted sequence  $faniradon(\delta_2)$  ranges from  $-2 * \delta_{\max}$  to  $2 * \delta_{\max}$ .
3. Create the zero matrix (M) (same size as that of the original matrix) with origin in the middle. The set of all polar co-ordinates  $(r, \phi)$  for the constant  $\delta_2$  are chosen and are filled up with  $faniradon(\delta_2)$  computed for the particular  $\delta_2$ . This is repeated for all values of  $\delta_2$ .
4. Compute  $L = 1/C_2^2$  matrix for the complete range of  $r$  and  $\phi$  as described in step 5.

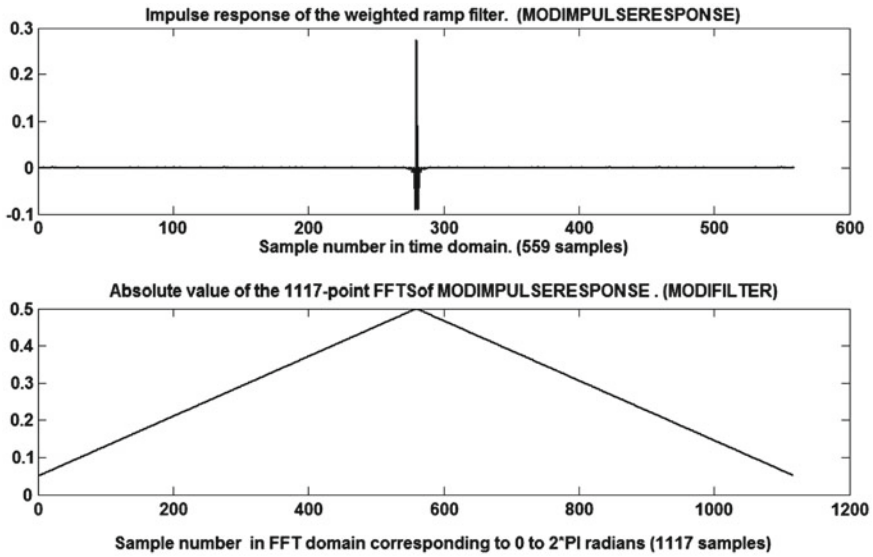


**Fig. 1.16** Fan-beam radon transformation of the Image A (refer Fig. 1.1) with  $C = 300$ ,  $\beta$  resolution =  $1.0028^\circ$  and  $\delta$  resolution =  $0.005$  radians

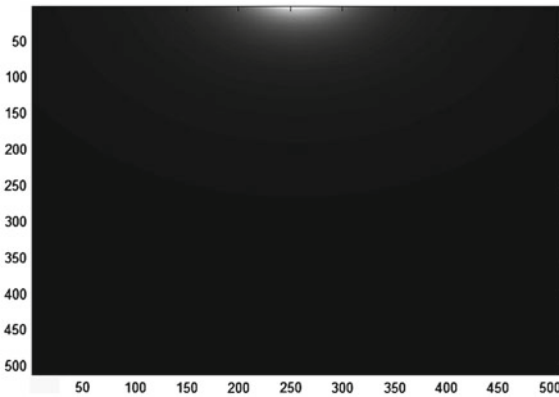
5. Create the matrix  $L$  exactly same as that of the size  $M$  with origin in the middle. Arbitrary  $(x, y)$  position of the matrix  $L$  is filled up with  $\frac{1}{x^2+(y-C)^2}$ .
6. Compute  $M1(x, y) = M(x, y)L(x, y)$ .
7. Rotate the obtained matrix  $M1$  in the clock wise direction by an angle  $\beta$ , say it as  $M1_\beta$ .
8. Compute  $M1_\beta$  for  $\beta$  ranging from  $0$  to  $2\pi$  with some resolution for  $\beta$ .
9. Thus the reconstructed image  $f(r, \phi)$  is computed as  $\sum_{\theta=0}^{\theta=2\pi} M1_\beta$ .

### 1.3.3 List of Figs. 1.12 to 1.25 in Terms of the Notations Used

1. Figure 1.12: Fan-beam projection geometry.
2. Figure 1.13–1.15: Relationship between the parallel beam and fan-beam projection geometry.
3. Figure 1.16:  $W(\delta, \beta)$  computed with  $\beta$  and  $\delta$  resolutions are respectively  $0.0175$  radians and  $0.005$  radians and  $C = 300$ .
4. Figure 1.17:  $\left(\frac{\delta_2}{\sin(\delta_2)}\right)^2 u(\delta_2)$  in time and frequency domain.
5. Figure 1.18: Normalized  $L$  matrix.
6. Figure 1.19: Illustration of obtaining convolution of  $\text{fanradon}(\delta_2)$  with  $\left(\frac{\delta_2}{\sin(\delta_2)}\right)^2 u(\delta_2)$  using DFT-IDFT (as described below).  
Linear convolution of two sequences  $x(\cdot)$  and  $h(\cdot)$  with  $K$  and  $L$  elements respectively is obtained as follows.
  - Pad the zeros in the sequence  $x(\cdot)$  and  $h(\cdot)$  at the end so that the length of both the sequences equal to  $K + L - 1$ .
  - Compute the dft of the padded sequences of  $x(\cdot)$  and  $h(\cdot)$  to obtain  $X(\cdot)$  and  $H(\cdot)$  respectively.



**Fig. 1.17** Impulse response of the weighted ramp filter (x-axis interpreted as  $\delta_2$  ranges from  $-\delta_{\max}$  to  $\delta_{\max}$ ) used in fan-beam projection and the corresponding spectrum



**Fig. 1.18** Normalized  $L$  matrix

- Compute idft of  $X.H$  to obtain the linear convolution.
7. Figure 1.20:  $M1_\beta$  for various  $\beta$  with  $\text{fanradon}(\delta_2) * \left(\frac{\delta_2}{\sin(\delta_2)}\right)^2 u(\delta_2) = \text{fanradon}(\delta_2)$  (i.e. without filtering).
  8. Figure 1.21:  $M1_\beta$  for various  $\beta$  (with filtering).
  9. Figure 1.22:  $\sum_{\beta=0}^{\beta=B} M1_\beta$  with  $\text{fanradon}(\delta_2) * \left(\frac{\delta_2}{\sin(\delta_2)}\right)^2 u(\delta_2) = \text{fanradon}(\delta_2)$  (i.e. without filtering) for various  $B$ .

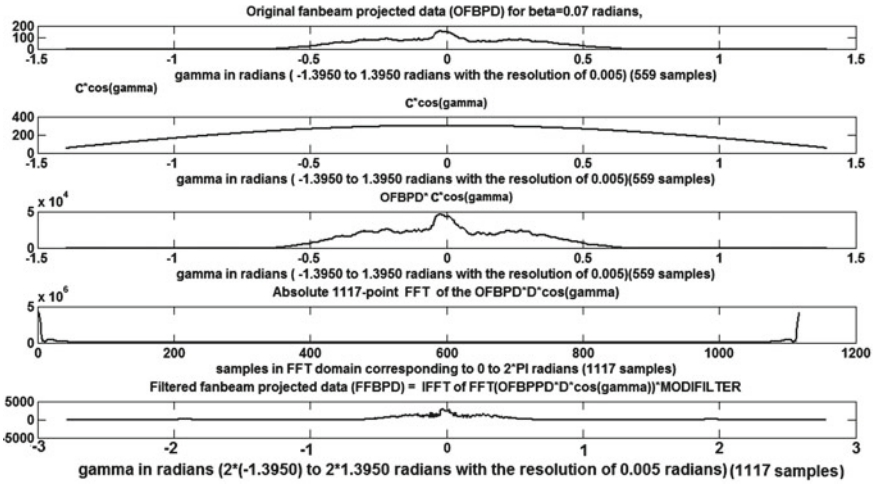


Fig. 1.19 Illustration of obtaining the filtered fan-beam projection data from the original fan-beam projection data for the angle  $\beta = 0.07$  radian

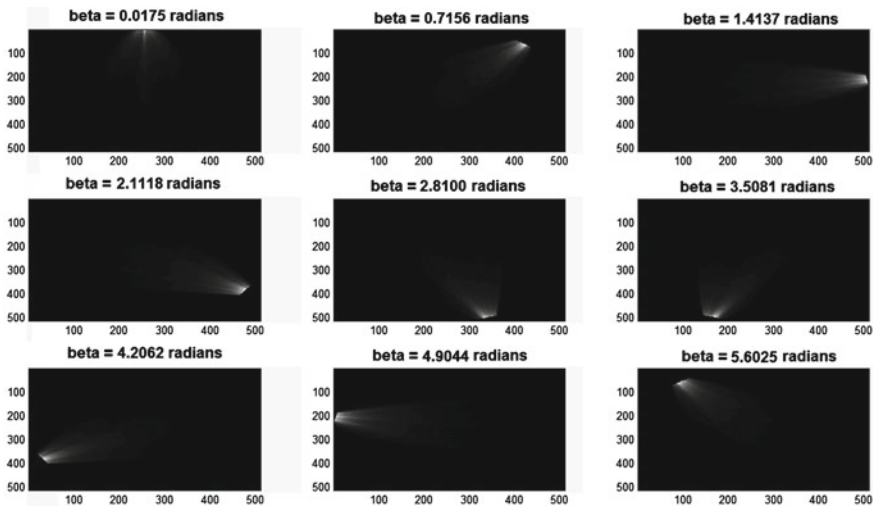
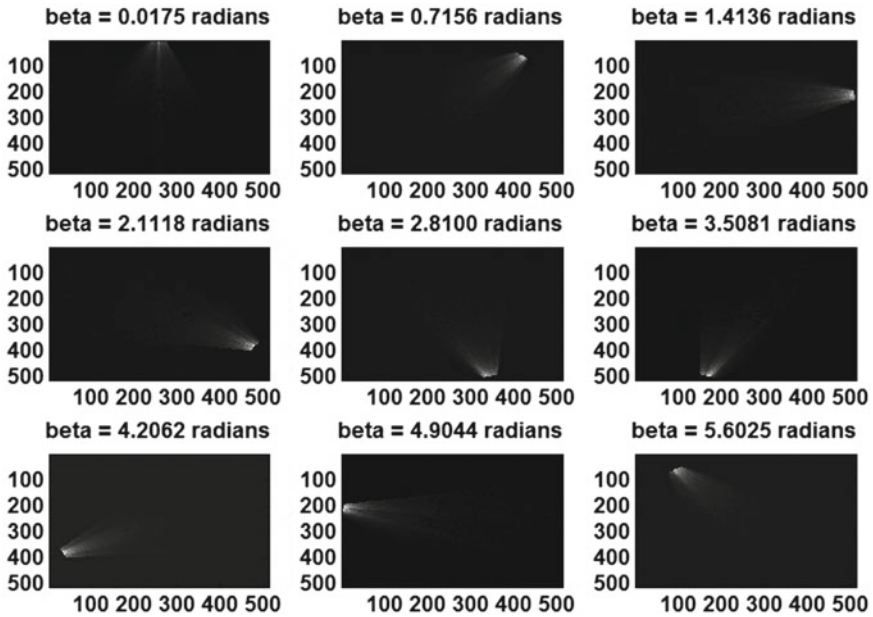
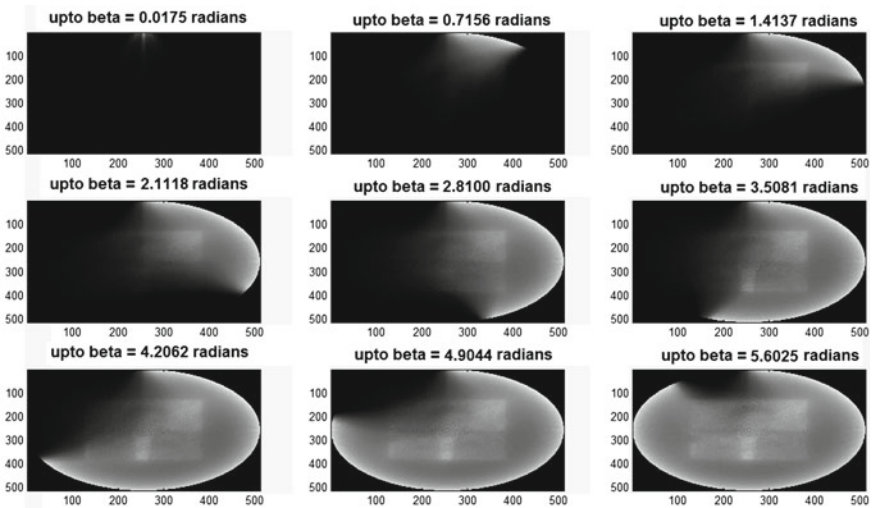


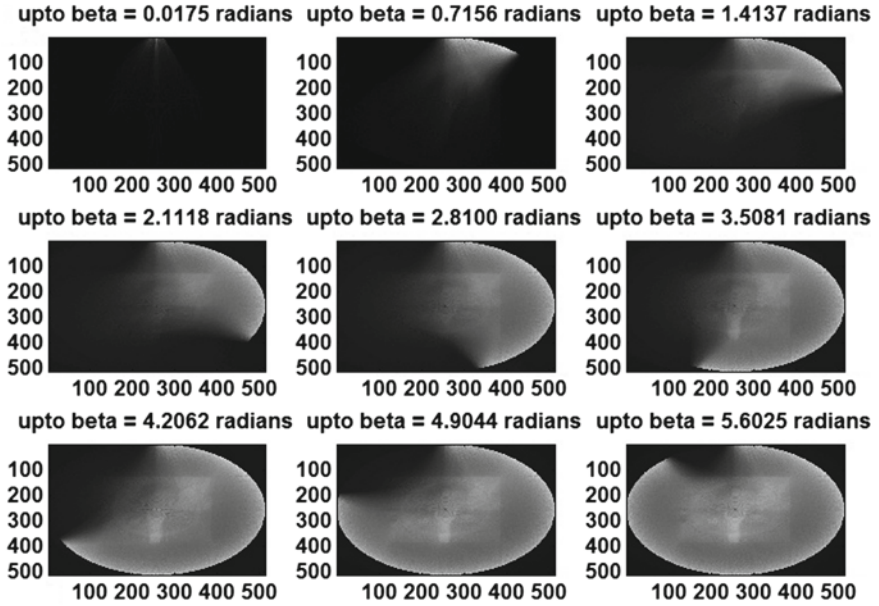
Fig. 1.20 Illustration of original fan-beam backprojected images obtained from the OFBPD for  $\beta = 0.0175, 0.7156, 1.4137, 2.1118, 2.8100, 3.5081, 4.2062, 4.9044, 5.6025$  radians. (Note that fan-beam reconstructed images are computed for all  $\beta$  ranging from 0 to 6.2657 with the resolution of 0.0175 radians)



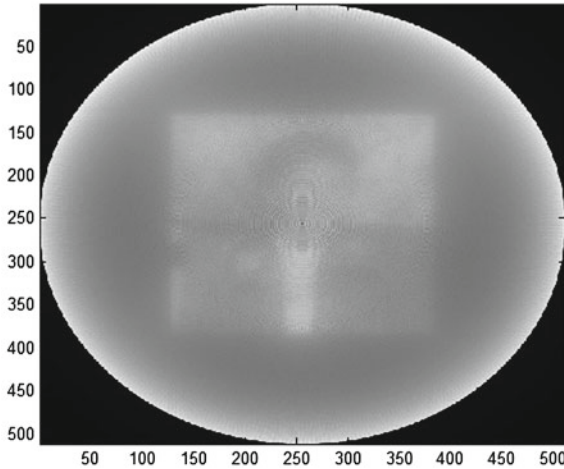
**Fig. 1.21** Illustration of filtered fan-beam backprojected images obtained from the FFBPD for  $\beta = 0.0175, 0.7156, 1.4137, 2.1118, 2.8100, 3.5081, 4.2062, 4.9044, 5.6025$  radians. (Note that fan-beam reconstructed images are computed for all  $\beta$  ranging from 0 to 6.2657 with the resolution of 0.0175 radians)



**Fig. 1.22** Illustration of reconstruction of the image obtained by cumulative summation of original fan-beam backprojected images

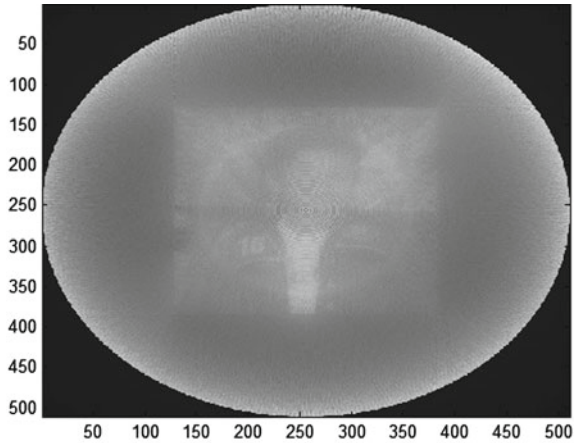


**Fig. 1.23** Illustration of reconstruction of the image obtained by cumulative summation of filtered fan-beam backprojected images



**Fig. 1.24** Final reconstructed image obtained from original fanbeam projection with  $C = 300$

**Fig. 1.25** Final reconstructed image obtained from filtered fanbeam projection with  $C = 300$



10. Figure 1.23:  $\sum_{\beta=0}^{\beta=B} M1_{\beta}$  for various  $B$  (with filtering).  
 11. Figure 1.24: Final fan-beam reconstructed image  $f(r, \phi)$  without filtering.  
 12. Figure 1.25: Final fan-beam reconstructed image  $f(r, \phi)$  with filtering.

### 1.3.3.1 Fanbeamprojection.m

```

%load the original image
load VASIGIMAGE
C=C(1:1:255,1:1:255);
C=[zeros(128,511);zeros(255,128) C zeros(255,128);
   zeros(128,511)];
C=double(C)/255;
figure
D=300;
maxiangle=atan(256/((D-255)+1));
B=zeros(511,511);
j=1;
range1=0:0.005:round(maxiangle/0.005)*0.005;
range2=sort(-1*range1);
ang=0:(2*pi)/359:2*pi;
angdeg=(ang/pi)*180;
for ang=angdeg(1:1:length(angdeg)-1)
t=0;
i=1;
A1=imrotate(C,ang,'crop');
%Computation of Beta
beta(j)=ang;
%Computation of delta
delta=[range2 range1(2:1:length(range1))];
for range=[range2 range1(2:1:length(range1))];

```



```

s=0;
%X and Y positions
X=-255:1:255;
Y=round((D-X)*tan(range));
Y1=-1*(Y+256)+512;
X1=-1*(X+256)+512;
COL{i}=[X1;Y1];
for u=1:1:511
if(X1(u)<=511&Y1(u)<=511&Y1(u)>0)
B(X1(u),Y1(u))=255;
s=s+A1(X1(u),Y1(u));
end
end
t(i)=s;
i=i+1;
end
fanbeam{j}=t;
t=0;
j=j+1;
end
figure
colormap(gray)
imagesc(B)
fanbeamprojection=fanbeam;
%Fan beam reconstruction
%4.Multiply with the modified ramp filter
%5.Weighted back projection
DATA=zeros(511,511);
RECONSTRUCTEDMATRIX=zeros(511,511);
%Computation of weight matrix L^(-2)
for i=1:1:511
    for j=1:1:511
        k=-i+512-256;
        l=-j+512-256;
L_MATRIX(j,i)=sqrt(k^2+(D-l)^2);
    end
end
L_MATRIX=L_MATRIX.^(-2)/max(max(L_MATRIX.^(-2)));
DATA=0;
figure
colormap(gray)
%Design of the weighted ramp filter
deltaterm=((delta.^2)./(sin(delta).^2));
deltaterm(280)=1;
MODIMPULSERESPONSE=(1/2)*fir2(558,[0:1/558:1],[0.1:0.9/558:1],
    hann(559)).*deltaterm;
%Fourier transformation of the weighted ramp filter
    after zero padding.
MODFILTER=fft(MODIMPULSERESPONSE,1117);
for k=1:1:359
RECONSTRUCTEDMATRIX=zeros(511,511);
%Computation of the Modified fanbeam projection
fanbeamprojection_modified1{k}=(D*cos(delta)).

```

```

                                *fanbeamprojection{k};
%Fourier transformation of the modified fan beam projection
after zero padding
fanbeamprojection_modified2{k}=fft(fanbeamprojection_
                                modified1{k},1117);
%Multiplication with the fourier transformation
of the weighted ramp
%filter after zero padding
%Computation of inverse fourier transformation
fanbeamprojection_modified3{k}=ifft(fanbeamprojection_
                                modified2{k}.*MODFILTER);
%Backprojection in the fanbeam structure
l=1;
for range=(-2*maxiangle):(4*maxiangle)/1116:(2*maxiangle)
X=-255:1:255;
Y=round((D-X)*tan(range));
Y1=-1*(Y+256)+512;
X1=-1*(X+256)+512;
for u=1:1:511
if((X1(u)<=511)&(Y1(u)<=511)&(Y1(u)>0))
RECONSTRUCTEDMATRIX(X1(u),Y1(u))=fanbeamprojection_
                                modified3{k}(l);

end
end
l=l+1;
end
%Multiplication with the weight matrix
RECONSTRUCTEDMATRIX=RECONSTRUCTEDMATRIX.*L_MATRIX;
%Obtaining the reconstructed matrix for beta(k)
RECONSTRUCTEDMATRIX=imrotate(RECONSTRUCTEDMATRIX,
                                -beta(k),'crop');
imagesc(RECONSTRUCTEDMATRIX)
pause(0.5)
DATA=DATA+RECONSTRUCTEDMATRIX;
end
figure
colormap(gray(256))
imagesc(DATA)

```

# Chapter 2

## Magnetic Resonance Imaging

### 2.1 Bloch Equation

The concept of MRI physics is described by the Bloch equations. Consider the weak magnetic field  $\vec{M}(t)$  kept at an angle  $\alpha$  (in the anticlockwise direction) with the strong magnetic field  $B(t)$  (which is kept in the  $z$ -direction as shown in the Fig. 2.1). The interaction between these magnetic fields end up with the torque (The rate of change of angular momentum  $\vec{J}(t)$  is the torque) on the weaker magnetic field  $\vec{B}(t)$  as mentioned in the Eq. (2.1).

$$\frac{d\vec{J}(t)}{dt} = \vec{M}(t) \times \vec{B}(t) \tag{2.1}$$

Note that the magnetic moment is proportional to the angular momentum (i.e)  $\vec{M}(t) = \gamma \vec{J}(t) \Rightarrow \frac{d\vec{M}(t)}{dt} = \gamma \vec{M}(t) \times \vec{B}(t)$ , where  $\gamma$  is gyromagnetic ratio of the magnetic moment  $\vec{M}(t)$ .

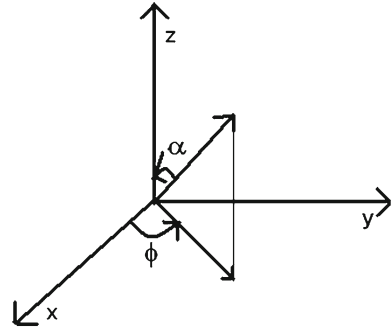
Let  $\vec{M}(t) = M_x(t)\hat{i} + M_y(t)\hat{j} + M_z(t)\hat{k}$  and  $\vec{B}(t) = B_x(t)\hat{i} + B_y(t)\hat{j} + B_z(t)\hat{k}$  with  $B_x(0) = 0, B_y(0) = 0$  and  $B_z(0) = B_0$

$$\Rightarrow \vec{M}(t) \times \vec{B}(t) = \begin{bmatrix} i & j & k \\ M_x(t) & M_y(t) & M_z(t) \\ 0 & 0 & B_0 \end{bmatrix}$$

This further implies,

$$\frac{dM_x(t)}{dt} = \gamma M_y(t) B_0 \tag{2.2}$$

**Fig. 2.1** Co-ordinate system illustrating Bloch equation



$$\frac{dM_y(t)}{dt} = -\gamma M_x(t)B_0 \quad (2.3)$$

$$\frac{dM_z(t)}{dt} = 0 \quad (2.4)$$

Let the projection of the initial magnetic moment  $\vec{M}(0)$  with magnitude  $M_0$  kept at an angle  $\alpha$  with the magnetic moment  $\vec{B}(t) = B_z(t)\hat{k} = B_0\hat{k}$  on the  $XY$ -plane is the vector with magnitude  $M_{xy}(0) = M_0 \sin(\alpha)$  and it makes an angle  $\phi$  (in the anti-clock wise direction) with the  $x$ -axis.

Note that the initial values of the magnetic moment  $\vec{M}(t)$  (with initial magnitude  $M_0$ ) projected on the three co-ordinates are mentioned as follows.

$$M_x(0) = M_{xy}(0) \cos(\phi) = M_0 \sin(\alpha) \cos(\phi) \quad (2.5)$$

$$M_y(0) = M_{xy}(0) \sin(\phi) = M_0 \sin(\alpha) \sin(\phi) \quad (2.6)$$

$$M_z(0) = M_0 \cos(\alpha) \quad (2.7)$$

To solve the Eq. (2.1), we assign  $M_{xy}(t) = M_x + M_y j$ , where  $j = \sqrt{-1}$ . Rewriting jointly the Eqs. (2.2) and (2.3), we get

$$\begin{aligned} \frac{dM_x(t)}{dt} + j \frac{dM_y(t)}{dt} &= \gamma M_y(t)B_z(t) - j\gamma M_x(t)B_z(t) \\ \Rightarrow \frac{dM_{xy}(t)}{dt} &= -j\gamma B_z(t)M_{xy}(t) \end{aligned}$$

Note that  $B_z(t)$  is constant and is represented as  $B_0$ .

$$\Rightarrow M_{xy}(t) = Ke^{-j\gamma B_0 t}$$

Applying the initial conditions (refer (2.5)–(2.7))  $M_{xy}(0) = M_0 \sin(\alpha) \cos(\phi) + jM_0 \sin(\alpha) \sin(\phi)$ , we get

$$\Rightarrow M_{xy}(t) = (M_0 \sin(\alpha) \cos(\phi) + jM_0 \sin(\alpha) \sin(\phi))e^{-j\gamma B_0 t}$$

$$\Rightarrow M_{xy}(t) = M_0 \sin(\alpha) e^{j\phi} e^{-j\gamma B_0 t}$$

$$\Rightarrow M_x(t) = M_0 \sin(\alpha) \cos(\phi - \gamma B_0 t) = M_0 \sin(\alpha) \cos(-\gamma B_0 t + \phi) \quad (2.8)$$

$$M_y(t) = M_0 \sin(\alpha) \sin(\phi - \gamma B_0 t) = M_0 \sin(\alpha) \sin(-\gamma B_0 t + \phi) \quad (2.9)$$

$$M_z(t) = M_0 \cos(\alpha) \quad (2.10)$$

## 2.2 Comment on the Equations 2.8–2.10

When the weak initial magnetic moment  $\vec{M}(0)$  with magnitude  $M_0$  is kept at an angle  $\alpha$  with the strong constant magnetic moment  $\vec{B}(t) = B_z(t)\hat{k} = B_0\hat{k}$ , due to bloch equation, magnetic moment in the  $z$ -direction remains unchanged. But the magnetic moment in the  $x$ -direction and the  $y$ -direction oscillates with the angular frequency of  $\gamma B_0$  radians or  $\frac{\gamma B_0}{2\pi}$  Hz with maximum amplitude  $M_0 \sin(\alpha)$ . Thus at any particular time instant, the magnitude of the resultant magnetic moment on the  $X$ - $Y$  plane is constant and is equal to  $M_0 \sin(\alpha)$ . Also note that at any particular time instant  $t$ , the resultant magnetic moment on the  $XY$ -plane is making an angle with magnitude  $(-\gamma B_0 t + \phi)$  with the  $x$ -axis measured in the anti-clock wise direction. As time goes, the magnitude of the angle is increasing. This implies the resultant vector on the  $XY$ -plane rotates in the anti-clock wise direction (when viewed in the  $z$  direction) with the frequency  $\gamma B_0$ . This frequency is called larmor frequency in radians and it is computed as  $\frac{\gamma B_0}{2\pi}$  in Hz. For the constant strong magnetic moment  $B_0$ , the larmor frequency purely depends on the gyromagnetic ratio of the magnetic moment  $\vec{B}(t)$ . Note that the magnitude of the resultant magnetic moment in the  $XY$ -plane (transverse plane) is directly proportional to the angle  $\alpha$ . Note: Clock wise direction is identified with respect to the view point in the direction of  $-z$  axis (refer Fig. 2.1).

## 2.3 The Larmor Frequency and the Tip Angle $\alpha$

In general, resultant magnetic moment (without external strong magnetic moment) obtained in the macroscopic level in the human body is zero. When the human body is kept under the constant strong magnetic moment of  $B_0$  in the  $z$ -direction. The resultant magnetic moment in the macroscopic level is aligned to the direction of the external strong magnetic moment (i.e)  $z$ -direction. When it is disturbed to bring the resultant magnetic moment to make an angle  $\alpha$  (measured anti-clock-wise direction) with the  $z$ -axis, there exists the resultant anti-clock-wise rotating magnetic moment in the transverse plane (due to bloch equations), that rotates with the frequency 42.58 Mhz, when  $B_0$  is 1 Tesla.

### 2.3.1 Disturbance to obtain Non-Zero $\alpha$ Value

The external field (apart from the strong constant magnetic moment  $B_0$ ) is applied for the short duration ( $\tau$ ) in such a way that the resultant magnetic moment  $\vec{E}(t)$  is rotating exactly with the larmor frequency of the magnetic moment  $\vec{M}(t)$  to be disturbed. It is noted that the macro magnetic moment  $\vec{M}(t)$  obtained using the hydrogen atoms that are aligned in the  $z$ -direction due to the availability of strong magnetic moment  $B_0$  in the  $z$ -direction. The interaction between the magnetic moment  $\vec{M}(t)$  aligned in the  $z$ -direction with magnitude  $M_0$  and the rotating magnetic moment  $\vec{E}(t)$  on the transverse plane is described by the bloch equations as described below. The strength of the magnetic moment  $\vec{E}(t)$  is strong compared with the natural magnetic moment  $\vec{M}(t)$  available in the human body that are aligned initially in the  $z$ -direction. Rewriting the bloch equation using  $\vec{M}(t)$  and  $\vec{E}(t)$ , we get the following.

$$\frac{d\vec{J}(t)}{dt} = \vec{M}(t) \times \vec{E}(t) \quad (2.11)$$

$$\Rightarrow \vec{M}(t) \times \vec{E}(t) = \begin{bmatrix} i & j & k \\ 0 & 0 & M_z(t) \\ E_x(t) & E_y(t) & 0 \end{bmatrix} \Rightarrow$$

$$\frac{dM_x(t)}{dt} = \gamma E_y(t) M_0 = \gamma E_0 \cos(-\gamma B_0 t + \theta) M_0 \quad (2.12)$$

$$\frac{dM_y(t)}{dt} = \gamma E_x(t) M_0 = \gamma E_0 \sin(-\gamma B_0 t + \theta) M_0 \quad (2.13)$$

$$\frac{dM_z(t)}{dt} = 0 \quad (2.14)$$

Solving the Eqs. (2.12)–(2.14) as described in the Sect. 2.1, we still get the resultant magnetic moment  $\vec{M}(t)$  lies only in the  $z$ -direction. It is noted from the equations that the transverse magnetic moment is zero due to the initial conditions  $M_x(0) = 0$ ,  $M_y(0) = 0$ .

But in practice, due to the external field, there is the disturbance in the resultant magnetic moment and there exist very low magnitude  $M_x$  and  $M_y$  component that rotates in the larmor frequency due to the existence of strong field  $B_0$  as described in the Sect. 2.1. Now consider the interaction between the magnetic fields  $\vec{E}(t)$  (which has  $E_x(t)$  and  $E_y(t)$  components) and  $\vec{B}(t)$  (which has  $B_z(t) = B_0$  component) on the magnetic moment  $\vec{M}(t)$  which have all the three components.

$$\frac{d\vec{M}(t)}{dt} = \gamma \vec{M}(t) \times (\vec{E}(t) + \vec{B}(t)) \quad (2.15)$$

The resultant magnetic moment  $\vec{M}(t)$  depends upon the first term and the second term of the RHS of the (2.16) independently. The resultant  $\vec{M}(t)$  due to the second term ends up with  $M_x(t)$ ,  $M_y(t)$  and  $M_z(t)$  components as described in the Eqs. (2.9)–(2.11). Note that the  $z$ -component of the resultant vector is constant due to the second term. Now the magnetic moment  $\vec{M}(t)$  due to the first term is obtained as follows. Rewriting (2.15) with only first term of the RHS as

$$\frac{d\vec{M}(t)}{dt} = \gamma \vec{M}(t) \times \vec{E}(t) \quad (2.16)$$

$$\Rightarrow \vec{M}(t) \times \vec{E}(t) = \begin{bmatrix} i & j & k \\ M_x(t) & M_y(t) & M_z(t) \\ E_x(t) & E_y(t) & 0 \end{bmatrix} \Rightarrow$$

$$\frac{dM_x(t)}{dt} = -\gamma E_y(t)M_z(t) = -\gamma E_0 \sin(-\gamma B_0 t + \theta)M_0 \cos(\alpha) \quad (2.17)$$

$$\frac{dM_y(t)}{dt} = \gamma E_x(t)M_z(t) = \gamma E_0 \cos(-\gamma B_0 t + \theta)M_0 \cos(\alpha) \quad (2.18)$$

$$\frac{dM_z(t)}{dt} = \gamma (M_x(t)E_y(t) - M_y(t)E_x(t)) \quad (2.19)$$

Solving the Eqs. (2.17)–(2.19) as described in Sect. 2.1, we get the following.

$$\begin{aligned} \frac{dM_x(t)}{dt} + j \frac{dM_y(t)}{dt} &= \gamma E_0 M_0 \cos(\alpha) (-\sin(-\gamma B_0 t + \theta) \\ &\quad + j \cos(-\gamma B_0 t + \theta)) \end{aligned} \quad (2.20)$$

$$\Rightarrow \frac{dM_{xy}(t)}{dt} = j\gamma E_0 M_0 \cos(\alpha) e^{j(-\gamma B_0 t + \theta)} \quad (2.21)$$

$$\Rightarrow M_{xy}(t) = \frac{-E_0 M_0 \cos(\alpha)}{B_0} e^{j(-\gamma B_0 t + \theta)} K,$$

where  $K$  is the constant.

As  $M_{xy}(0) = M_0 \sin(\alpha) \cos(\phi) + jM_0 \sin(\alpha) \sin(\phi) = M_0 \sin(\alpha) e^{j\phi}$  we get  $K$  as follows.

$$\begin{aligned} M_{xy}(0) &= \frac{-E_0 M_0 \cos(\alpha)}{B_0} e^{j\theta} K \\ \Rightarrow \frac{-E_0 M_0 \cos(\alpha)}{B_0} \cos(\theta) K &= M_0 \sin(\alpha) \cos(\phi) \end{aligned}$$

Also,

$$\begin{aligned}
\frac{-E_0 M_0 \cos(\alpha)}{B_0} \sin(\theta) K &= M_0 \sin(\alpha) \sin(\phi) \\
\Rightarrow K &= \frac{-B_0}{E_0} \tan(\alpha) e^{j(\phi-\theta)} \\
\Rightarrow M_{xy}(t) &= \frac{-E_0 M_0 \cos(\alpha)}{B_0} e^{j(-\gamma B_0 t + \theta)} \frac{-B_0}{E_0} \tan(\alpha) e^{j(\phi-\theta)} \\
\Rightarrow M_{xy}(t) &= M_0 \sin(\alpha) e^{j(-\gamma B_0 t + \phi)}
\end{aligned}$$

Note that, the transverse component is not changed due to the external field  $\vec{E}(t)$ . What we achieved is that the transverse magnetic moment  $M_{xy}(t)$  due to the external field  $\vec{E}(t)$  is in phase as that of the transverse component obtained using the static magnetic field  $B_0$ . The resultant transverse magnetic moment due to  $B_0$  and  $\vec{E}(t)$  is given as

$$M_{xy}(t) = 2M_0 \sin(\alpha) e^{j(-\gamma B_0 t + \phi)} \quad (2.22)$$

The effect of the external magnetic moment  $\vec{E}(t)$  on the  $z$ -component of the magnetic moment  $\vec{M}(t)$  is obtained by solving (2.19) as shown below.

$$\begin{aligned}
\frac{dM_z(t)}{dt} &= \gamma(M_x(t)E_y(t) - M_y(t)E_x(t)) \\
&= \gamma M_0 E_0 \sin(\alpha) (\sin(-\gamma B_0 t + \theta) \cos(-\gamma B_0 t + \phi) \\
&\quad - \cos(-\gamma B_0 t + \theta) \sin(-\gamma B_0 t + \phi)) \\
&= \gamma M_0 \sin(\alpha) E_0 \sin(-\gamma B_0 t + \theta - (-\gamma B_0 t + \phi))
\end{aligned}$$

Thus

$$\frac{dM_z(t)}{dt} = \gamma M_0 \sin(\alpha) E_0 \sin(\theta - \phi) \quad (2.23)$$

$$\Rightarrow M_z(t) = \gamma M_0 \sin(\alpha) E_0 \sin(\theta - \phi) t + M_z(0) \quad (2.24)$$

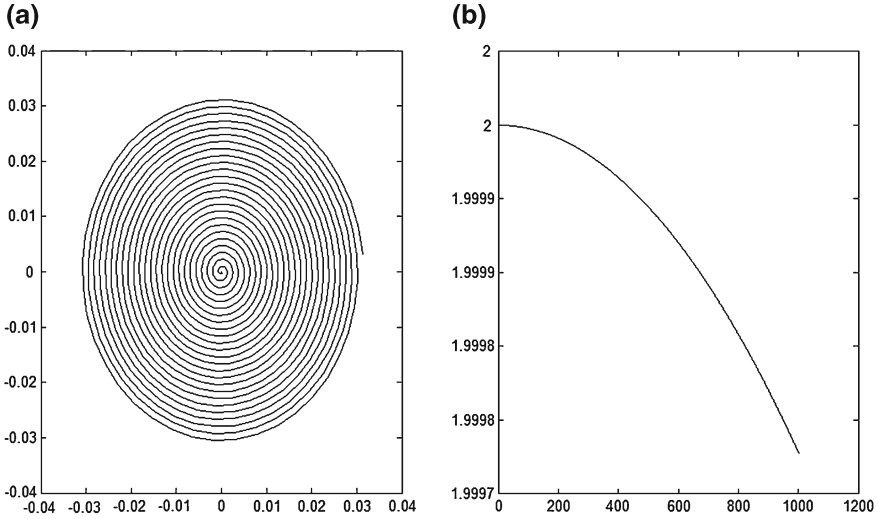
Applying the initial condition  $M_z(0) = M_0 \cos(\alpha)$  in (2.24), we get  $M_z(t) = \gamma M_0 \sin(\alpha) E_0 \sin(\theta - \phi) t + M_0 \cos(\alpha)$ . Recall that the  $z$ -component due to  $B_0$  is  $M_0 \cos(\alpha)$  and hence resultant  $z$ -component is obtained as follows

$$M_z(t) = \gamma M_0 \sin(\alpha) E_0 \sin(\theta - \phi) t + 2M_0 \cos(\alpha) \quad (2.25)$$

It is also noted that the resultant  $z$ -component with external field  $B_1$  (instead of  $B_0$ ) is given as

$$M_z(t) = \gamma M_0 \sin(\alpha) E_0 \sin(-\gamma(B_1 - B_0) + \theta - \phi) t + 2M_0 \cos(\alpha) \quad (2.26)$$





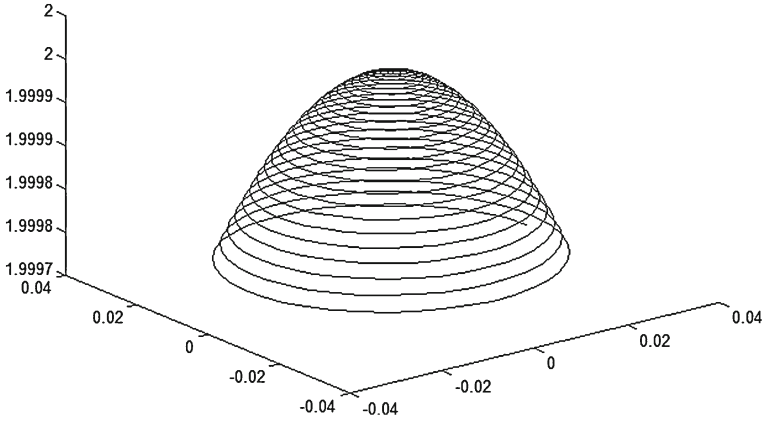
**Fig. 2.2** **a** Trace of the transverse component of the resultant magnetic moment for the duration of  $t \frac{\pi}{200}$  (refer Sect. 2.3.2) **b** Illustrating how the  $z$ -component of the magnetic moment is decreasing as time increases

**2.3.2 Observation on (2.22) and (2.25)**

1. Magnitude of the transverse component increases with  $\alpha$  value (refer 2.22).
2.  $z$ -component decreases with incremental change in the time  $t$  (refer 2.25). This is equivalently viewed as the effective increase in  $\alpha$  value.
3. This helps in further increasing the magnitude of the transverse component.
4. Note that the resultant magnetic moment is rotating with the larmor frequency.
5. Thus we can imagine that the rotating magnetic moment is moving towards the  $XY$ -plane. This is equivalently viewed as the spiral trajectory traced by the transverse component of the  $\vec{M}(t)$  on the  $XY$ -plane, while  $z$ -component of the  $\vec{M}(t)$  is decreasing along the  $z$ -direction (refer Fig. 2.2).
6. This in further looks like the helical movement of the resultant magnetic moment as shown in the Fig. 2.3.

Thus the external magnetic moment  $\vec{E}(t)$  which is applied for the duration  $T_{\pi/2}$  helps in bringing the magnetic moment  $\vec{M}(t)$  to the  $XY$ -plane. Note that the  $T_{\pi/2}$  is also the time duration to make the resultant  $\alpha$  value  $\frac{\pi}{2}$ . It is computed as described below.

The magnitude of the magnetic moment making an angle  $\alpha$  with the  $z$ -axis be  $M_0$ . The  $z$ -component of the magnetic moment is given as  $M_0 \cos(\alpha)$ . The rate at which  $z$ -component of the magnetic moment is decreasing is proportional to the rate at which the angle  $\alpha$  is changing (refer 2.25). This implies



**Fig. 2.3** Trace of the resultant magnetic moment in 3D for the duration of  $t_{\frac{\pi}{200}}$  (refer Sect. 2.3.2)

$$-M_0 \sin(\alpha) \frac{d\alpha}{dt} = \gamma M_0 \sin(\alpha) E_0 \sin(\theta - \phi) \quad (2.27)$$

$$\Rightarrow \frac{d\alpha}{dt} = -\gamma E_0 \sin(\theta - \phi) \quad (2.28)$$

$$\Rightarrow \alpha(t) = -\int_0^t \gamma E_0 \sin(\theta - \phi) dt \quad (2.29)$$

If the external field is  $B_1$ , then  $\alpha(t)$  is computed as follows (refer 2.26).

$$\alpha(t) = -\int_0^t \gamma E_0 \sin(-\gamma(B_1 - B_0) + \theta - \phi) dt \quad (2.30)$$

### 2.3.2.1 resmagneticmoment.m

```

gamma=2*pi*42.58*10^6;
E0=0.0001;
M0=1;
B0=1;
phi=0.1;
theta=0.2;
x=[];
y=[];
z=[];
duration=5.8713*10^(-5);
for t=0:(duration/100000):(duration/100)
alpha1= E0*gamma*t;
x=[x 2*M0*sin(alpha1)*cos(-gamma*B0*t+phi)];

```

```

y=[y 2*M0*sin(alpha1)*sin(-gamma*B0*t+phi)];
z=[z gamma*M0*sin(alpha1)*E0*sin(theta-phi)*t+2*M0*cos(alpha1)];
end
figure
plot3(x,y,z)
figure
subplot(1,2,1)
plot(x,y)
subplot(1,2,2)
plot(z)

```

## 2.4 Trick on MRI

The external magnetic moment  $\vec{E}(t)$  is applied for the duration  $T_{\pi/2}$  to bring the resultant magnetic moment to the transverse plane as described in the Sect. 2.3. After that, when  $\vec{E}(t)$  is removed, the resultant magnetic moment has to rotate with constant magnitude in the  $XY$ -plane with the larmor frequency. But in nature, transverse component decreases and reaches zero after some time. This is called spin relaxation. This is due to the spin–spin interactions between the micro-level magnetic moments available in the human body. The time required to obtain  $(1/e)$  times the initial value of the transverse component after the removal of the external magnetic moment (represented as  $T_2$ ) depends on the characteristics of the tissues of human body. The resultant transverse magnetic moment during relaxation (free induction decay (FID)) is recorded using the receiver antenna. This is used to obtain  $T_2$  MRI and proton-density MRI images.

In the same fashion, longitudinal component gradually increases and attains the maximum value. This is due to spin-lattice interactions in the micro level magnetic moments of the human body. The rate at which longitudinal component reaches the maximum value is described by the time constant  $T_1$  (depends on the characteristics of the tissues of the human body). Usually  $T_1 \gg T_2$ . After sufficient time (to nullify the influence of existing transverse component), longitudinal component is flipped to the transverse component and the corresponding FID is measured. This is used to obtain the  $T_1$  MRI image. Note that in all the cases ( $T_1$ ,  $T_2$  and proton-density) MRI images, the receiver records only the transverse components of the magnetic field during relaxation. The complete description is given in Sect. 2.8.

## 2.5 Selecting the Human Slice and the Corresponding External RF Pulse

When the external RF frequency is same as that of the larmor frequency, we are able to get the transverse component of the magnetic field. When the complete human body is kept under the identical strong magnetic moment  $B_0$ , the recorded

FID signal corresponds to the complete human body. But we need to get the image of the particular slice. This is obtained using the concept of Gradient. Let us assume that we need to image the particular slice of the human body along the  $z$ -axis. We apply the gradient magnetic moment such that the  $z$ -component of the static magnetic field  $B_z(z)$  is the function of  $z$ . The resultant  $z$ -component of the magnetic moment is given as  $B_z(z) = G_z z + K$ , where  $G_z$  is the  $z$ -gradient and  $K$  is the constant. The constant is chosen such that at the required point  $z$ , the magnitude of the magnetic moment is  $B_0$ . (i.e)

$$B_z(z) = G_z(z - \bar{z}) + B_0 \quad (2.31)$$

Recollect that the external field used to disturb the original magnetic moment  $M(t)$  (to obtain non-zero alpha) which are kept under the strong magnetic field  $B_0$  in the  $z$ -axis is given as  $E_x(t) = -\gamma E_0 \sin(-\gamma B_0 t + \theta)$  and  $E_y(t) = -\gamma E_0 \cos(-\gamma B_0 t + \theta)$  (refer 2.12 and 2.13). We can still use the same external field for disturbance. But it controls the slice corresponding to the frequency  $\gamma B_0$ . In practice it is difficult to generate such signals. so the alternate technique is to choose the slice with pre-determined thickness. Let us assume, we need to image the slice corresponds to the magnetic field ranging between  $B_{z_1}$  and  $B_{z_2}$ , where  $\bar{z} = \frac{(z_1 + z_2)}{2}$ . In this case, the external magnetic field is chosen such that  $T_\alpha$  for all the magnetic moments of the chosen slice is identical. It is given as follows.

$$E_x(t) = -A \Delta v \gamma \text{sinc}(\Delta v t) \sin(-\gamma B_0 t + \theta) \quad (2.32)$$

$$E_y(t) = -A \Delta v \gamma \text{sinc}(\Delta v t) \cos(-\gamma B_0 t + \theta) \quad (2.33)$$

It is noted that the external field is having only transverse components. Also note that the  $\Delta v$  is the bandwidth in Hz, which is computed as  $\frac{\gamma(B_{z_1} - B_{z_2})}{2\pi}$ . Using (2.31) we obtain,  $\Delta v = \frac{\gamma G_z (z_1 - z_2)}{2\pi}$ . It is noted  $\Delta v \ll \frac{B_0 \gamma}{2\pi}$ . Using this condition, Bloch equations with external fields (refer Sect. 2.3.2 and appendix 1) is solved to obtain the following equation for  $\alpha(t)$ .

$$\alpha(t, z) = - \int_0^t A \Delta v \gamma \text{sinc}(\Delta v t) \sin(-\gamma(B_z(z) - B_0) + \theta - \phi) dt \quad (2.34)$$

Note that the time instant at which  $t = 0$  is the starting time at which the external field is applied. In (2.28), the amplitude (envelope) of the external field is constant. But in this case, the starting time is properly chosen as the envelope is the  $A \Delta v \text{sinc}(\Delta v t)$  function. The sinc function is given as  $\text{sinc}(\Delta v t) = \frac{\sin \pi \Delta v t}{\pi \Delta v t}$  and is maximum at  $t = 0$ .

Suppose let us assume the case we are applying the  $A \Delta v \text{sinc}(\Delta v t)$  pulse for the duration from  $-\infty$  to  $\infty$  (which is not practical). The obtained  $\alpha$  values as the function of  $z$  (refer Sect. 2.3.2) is obtained as follows.

$$\alpha(z) = - \int_{-\infty}^{\infty} A \Delta v \gamma \text{sinc}(\Delta v t) \sin(-\gamma G_z(z - \bar{z})t + \theta - \phi) dt \quad (2.35)$$

$$\Rightarrow \alpha(z) = \text{Im} \left( - \int_{-\infty}^{\infty} A \Delta v \gamma \text{sinc}(\Delta v t) e^{j(-\gamma G_z(z - \bar{z})t + \theta - \phi)} dt \right) \quad (2.36)$$

Consider the following equation as the fourier transformation of the  $A \Delta v \gamma \sin c(\Delta v t) e^{j(\theta - \phi)}$  with frequency  $\gamma G_z(z - \bar{z})$  (in radians)

$$- \int_{-\infty}^{\infty} A \Delta v \gamma \text{sinc}(\Delta v t) e^{j(-\gamma G_z(z - \bar{z})t + \theta - \phi)} dt \quad (2.37)$$

Solving we get the following.

$$\alpha(z) = -A\gamma \sin(\theta - \phi) \text{rect} \left( \frac{\gamma G_z(z - \bar{z})}{2\pi \Delta v} \right) \quad (2.38)$$

$$\Rightarrow \alpha(z) = -A\gamma \sin(\theta - \phi) \text{rect} \left( \frac{(z - \bar{z})}{\Delta z} \right) \quad (2.39)$$

where  $\Delta z = \frac{2\pi \Delta v}{\gamma G_z}$

Thus the obtained  $\alpha$  for the entire slice is constant. But in practice the sinc envelope is not applied for the infinite duration. It is applied during the duration  $-\tau_p/2$  to  $\tau_p/2$ . This makes the  $\alpha(z)$  profile not perfectly flat. From signal processing, we understand that  $FT(\text{rect}(\frac{t}{\tau_p}) A \Delta v \gamma \text{sinc}(\Delta v t))$  computed with frequency  $\gamma G_z(z - \bar{z})$  is same as that of the convolution of  $FT(\text{rect}(\frac{t}{\tau_p}))$  (computed with frequency  $\gamma G_z(z - \bar{z})$ ) and  $FT(A \Delta v \gamma \text{sinc}(\Delta v t))$  (computed with frequency  $\gamma G_z(z - \bar{z})$ ). This implies the following expression for  $\alpha(z)$ .

$$\alpha(z) = -A\gamma \sin(\theta - \phi) \text{rect}\left(\frac{(z - \bar{z})}{\Delta z}\right) * \tau_p \text{sinc}(\tau_p \gamma G_z(z - \bar{z})) \quad (2.40)$$

Thus using the external field (2.32) and (2.33), we obtain almost the identical  $\alpha$  over the region of interest (slice region). Thus the particular slice of the human body along the  $z$ -axis is selected. Note that external field is applied over the duration  $-\tau_p/2$  to  $\tau_p/2$ .

At the end of time instant  $\tau_p/2$  (after acquiring required  $\alpha$  value),  $M_{xy}(t, z)$  is obtained as follows (refer (2.22)).

$$M_{xy}(\tau_p/2) = 2M_0 \sin(\alpha_{\tau_p/2}) e^{j(-\gamma B_z(z) \tau_p/2 + \phi)} \quad (2.41)$$

$$\Rightarrow M_{xy}(\tau_p/2) = 2M_0 \sin(\alpha_{\tau_p/2}) e^{j(-\gamma(G_z(z - \bar{z}) + B_0) \tau_p/2 + \phi)} \quad (2.42)$$

Note that the phase component of (2.42) varies with  $z$  as  $e^{j(-\gamma(G_z(z-\bar{z})\tau_p/2))}$ . To nullify this, negative  $z$ -gradient  $-G_z$  (along with the existent strong magnetic field  $B_0$ ) is applied. This is known as Refocussing gradient. Note that the external field (2.32) and (2.33) are removed. When the resultant magnetic moment is having non-zero  $\alpha$  and are kept with the strong magnetic field  $-(G_z(z-\bar{z})+B_0)$  for the time duration  $\tau_p/2$ , rotating transverse component (after  $\tau_p/2$ ) is obtained and is given as follows.

$$2M_0 \sin(\alpha_{\tau_p/2}) e^{j(-\gamma(G_z(z-\bar{z})+B_0)\tau_p/2+\phi)} e^{j(-\gamma(-G_z(z-\bar{z})+B_0)\tau_p/2)} \quad (2.43)$$

$$\Rightarrow 2M_0 \sin(\alpha_{\tau_p/2}) e^{j(-\gamma(B_0)\tau_p+\phi)} \quad (2.44)$$

Thus the resultant phase component is constant throughout the slice (not the function of  $z$ ). But note that there is still strong magnetic field  $B_0$  available in the  $z$ -axis. Hence transverse magnetic moments along the slice are having the same phase, having non-zero  $\alpha$  value and are under the constant magnetic field  $B_0$ . Hence transverse components of the magnetic field along the slice follows the equation as mentioned below.

$$2M_0 \sin(\alpha_{\tau_p/2}) e^{j(-\gamma(B_0)\tau_p+\phi)} e^{j(-\gamma(B_0)t)} \quad (2.45)$$

In (2.45),  $t = 0$  corresponds to  $\tau_p$  in the time scale  $t'$ , where  $t' = 0$  corresponds to the middle of the sinc pulse applied. As described in the Sect. 2.4, the resultant transverse component (refer 2.46) gradually decreases due to spin–spin interactions. This interactions start at the moment when there is non-zero  $\alpha$  value. so at time  $t = 0$  (middle of the RF pulse) itself, the transverse components decreases with time constant  $T_2$ . If there is no refocussing gradient and other externally disturbing fields, the disturbance in the transverse component is only due to spin–spin interaction, which is completely described by the time constant  $T_2$ . For instance, after applying refocussing gradient, the transverse component of the magnetic moment (including the effect of spin–spin relaxation) is given as follows.

$$2M_0 \sin(\alpha_{\tau_p/2}) e^{j(-\gamma(B_0)\tau_p+\phi)} e^{j(-\gamma(B_0)t)} e^{\frac{-t}{T_2(x,y)}} e^{\frac{-\tau_p}{T_2(x,y)}} \quad (2.46)$$

Note that  $T_2$  is the function of  $(x, y)$  due to different physical characteristics of the tissues. The transverse component of the signal is sampled at some time instant  $T_R$  (read-out time instant) depends on the factor  $T_2(x, y)$  and helps for  $T_2$ -MRI imaging technique.

### 2.5.1 Summary of the Section 2.5

1. Apply positive  $z$ -gradient  $G_z$  to select the slice of the human body along the  $z$ -axis.
2. RF pulse  $A\Delta v\gamma \text{sinc}(\Delta v t) \text{rect}(\frac{t}{\tau_p})$  is applied (Note that the duration is between  $-\tau_p/2$  and  $\tau_p/2$ ) to obtain the identical  $\alpha$  throughout the slice.

3. Negative  $z$ -gradient  $-G_z$  (refocussing gradient) (for the duration  $\tau_p/2$  to  $\tau_p$ ) is applied to achieve the identical strong magnetic field throughout the slice, by nullifying the phase introduced during RF excitation.
4. Step 3 helps in obtaining the transverse magnetic components within the selected slice (in all  $(x, y)$  co-ordinates) to rotate with the Larmor frequency with zero phase difference for a moment.
5. But the transverse component gradually decreases with time due to spin-spin interaction. This is known as relaxation. The rate at which the transverse components decrease depends on the location (with different tissue properties at  $(x, y)$ ) described by the time constant  $T_2$ .
6. Transverse components are assumed to start decaying from the time instant  $t = 0$ , which is the middle of the applied RF pulse.
7. We need to measure the transverse component during relaxation which acts as the first step to obtain MRI image as described in Sect. 2.6

## 2.6 Measurement of the Transverse Component Using the Receiver Antenna

In general, the transverse magnetic moment and the longitudinal magnetic moment during the readout phase are given as follows.

$$M_{xy}(x, y, t) = M_{xy}(x, y, 0^+) e^{j(-\gamma(B_0)t + \phi)} e^{-\frac{t}{T_2(x,y)}}$$

$$M_{xy}(x, y, 0^+) = M_z(x, y, 0^-) \sin(\alpha(x, y))$$

$$M_z(x, y, t) = M_z(x, y, 0^+) e^{-t/T_1} + (1 - e^{-t/T_1}) B_0$$

$$M_z(x, y, 0^+) = M_z(x, y, 0^-) \cos(\alpha(x, y))$$

To obtain the image that describes the  $T_{x,y}$  property of the sliced XY plane (selected slice plane), readout time should be chosen such that the  $\alpha(x, y)$  value must be constant throughout the plane. Rewriting the equations with constant  $\alpha(x, y)$  is as follows.

$$M_{xy}(x, y, t) = M_{xy}(x, y, 0^+) e^{j(-\gamma(B_0)t + \phi)} e^{-\frac{t}{T_2(x,y)}} \quad (2.47)$$

$$M_{xy}(x, y, 0^+) = M_z(x, y, 0^-) \sin(\alpha) \quad (2.48)$$

$$M_z(x, y, t) = M_z(x, y, 0^+) e^{-t/T_1} + (1 - e^{-t/T_1}) B_0 \quad (2.49)$$

$$M_z(x, y, 0^+) = M_z(x, y, 0^-) \cos(\alpha) \quad (2.50)$$

### 2.6.1 Observation on (2.47)–(2.50)

1. The time instant  $t = 0$  in (2.47) is the starting time at which the receiver starts receiving the signal. This is otherwise called as the starting time instance of the readout phase.
2. Note that the transverse component is rotating with identical larmour frequency at all  $(x,y)$  positions. This is achieved with the identical strong magnetic field (in the  $z$ -axis) throughout the slice.
3. The amplitude  $M_{xy}(x, y, 0^+)$  is the function of  $(x, y)$  as it involves the hidden term  $e^{\frac{-t}{T_2(x,y)}}$  from the time instance of the middle of the RF pulse.
4. Hence if the receiver is designed to receive the transverse component as the function of  $(x, y)$ , the image completely describes the  $T_2(x, y)$  characteristics of the sliced tissue.

### 2.6.2 Receiver to Receive the Transverse Component

The transverse component  $M(x, y, t)$  is represented as the vector  $[M_{x,t} M_{y,t}]$ . Usually  $M_{x,t}$  component is sensed as the voltage induced in the the receiver coil as described below. When the receiver coil is excited with the external source to generate the transverse magnetic moment represented as the vector  $[1 \ 0]$  and are kept in the transverse magnetic moment (to be sensed) represented as the vector  $[M_{x,t} M_{y,t}]$ , the voltage is induced in the coil as follows.

$$v(t) = -\frac{d}{dt} \int_x \int_y [M_{x,t} M_{y,t}] \cdot [1 \ 0] dx dy \quad (2.51)$$

The generalized expression for the transverse component of the magnetic moment is given as

$$M(x, y, t) = (M_r + jM_i)e^{-j(\gamma B_0 t - \phi)} \quad (2.52)$$

$$\Rightarrow v(t) = -\frac{d}{dt} \int_x \int_y [(M_r \cos(\gamma B_0 t - \phi) + M_i \sin(\gamma B_0 t - \phi))$$

$$(-M_r \sin(\gamma B_0 t - \phi) + M_i \cos(\gamma B_0 t - \phi))] \cdot [1 \ 0] dx dy \quad (2.53)$$

$$\Rightarrow v(t) = \gamma B_0 \int_x \int_y (M_r \sin(\gamma B_0 t - \phi) - M_i \cos(\gamma B_0 t - \phi)) dx dy \quad (2.54)$$

It is possible to obtain  $M_r$  and  $M_i$  as follows.



1. Multiply  $v(t)$  with  $\sin(\gamma B_0 t - \phi)$  and pass it through the low pass filter to obtain  $K \gamma B_0 \int_x \int_y M_r dx dy$  component, where  $K$  is the constant. Note that phase lock is also required.
2. Similarly, multiply  $v(t)$  with  $-\cos(\gamma B_0 t - \phi)$  and pass it through the low pass filter to obtain  $K \gamma B_0 \int_x \int_y M_i dx dy$  component.
3. Thus the complex number  $C \int_x \int_y (M_r + jM_i) dx dy$  can be stored in the computer as the complex number, where  $C$  is real constant.
4. In MRI imaging,  $M_r, vM_i$  are usually the function of time which are represented as  $M_r(t), M_i(t)$ . The frequency content of the signals  $M_r(t)$  and  $M_i(t)$  are comparatively very less when compared with the frequency  $\gamma B_0$ . Hence the same procedure (as described in 1 and 2) can be used to obtain the complex number  $C_1 \int_x \int_y (M_r(t) + jM_i(t)) dx dy$  as the function of time, where  $C_1$  is some arbitrary real constant.
5. Sampling the signal at any time instant gives the constant complex number, which can be stored in the computer.

## 2.7 Sampling the MRI Image in the Frequency Domain

We understand that the the receiver is capable of receiving the real and imaginary component of the signal  $s(t)$  (function of  $t$ ) mentioned in (2.55). Recall that the  $M_{xy}(x, y, 0^+)$  is the complex quantity.

$$s(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} M_{xy}(x, y, 0^+) e^{j(\phi)} e^{\frac{-t}{T_2(x,y)}} dx dy \quad (2.55)$$

Suppose that the external strong magnetic field (in the  $z$ -axis) is made as the function of  $x$  and  $y$  (i.e)  $B_z(x, y) = B_0 + xg_x + yg_y$  (Note that  $g_x$  and  $g_y$  are constants), we get

$$s(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} M_{xy}(x, y, 0^+) e^{j(-\gamma(B_z(x,y))t + \phi)} e^{\frac{-t}{T_2(x,y)}} dx dy \quad (2.56)$$

$$\Rightarrow s(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} M_{xy}(x, y, 0^+) e^{j(-\gamma(B_0 + xg_x + yg_y)t + \phi)} e^{\frac{-t}{T_2(x,y)}} dx dy \quad (2.57)$$

By varying the constants  $g_x g_y$  described by the variables  $G_x, G_y$  respectively, the same receiver is now capable of obtaining the real and imaginary component of the signal  $s(G_x, G_y, t)$ .

$$s(G_x, G_y, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} M_{xy}(x, y, 0^+) e^{j(-\gamma(xG_x + yG_y)t + \phi)} e^{\frac{-t}{T_2(x,y)}} dx dy \quad (2.58)$$

Sample the obtained complex signal at the middle of the readout time duration ( $T_{\text{readout}}/2$ ), which is the function of  $G_x$  and  $G_y$ . By vaying different values of  $G_x$  and  $G_y$  we obtain the real matrix  $R(G_x, G_y)$  and the imaginary matrix  $I(G_x, G_y)$ . Note that  $G_x$  and  $G_y$  ranges from the  $-G_{\text{max}}/2$  to  $G_{\text{max}}/2$  with the step increment of  $G_{\text{max}}/L$ , where  $L$  is the level number. Let us assume that the complete slice has to pictured with  $101 \times 101$  pixel resolution, then the  $G_x$  and  $G_y$  must have  $L = 101$ . This is equivalent to sampling the image in frequency domain.

$$C(G_x, G_y) = R(G_x, G_y) + jI(G_x, G_y) \quad (2.59)$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} M_{xy}(x, y, 0^+) e^{j(-\gamma(xG_x + yG_y)(T_{\text{readout}}/2) + \phi)} e^{\frac{-(T_{\text{readout}}/2)}{T_2(x,y)}} dx dy \quad (2.60)$$

Apply the linear map of the variable  $G_x$  to  $U$  and  $G_y$  to  $V$  as  $G_x = (G_{\text{max}}/(L - 1))U + (G_{\text{max}}/2)$  and  $G_y = (G_{\text{max}}/(L - 1))V - (G_{\text{max}}/2)$  and rewrtng the matrix  $C(G_x, G_y)$  as  $C1(U, V)$ , where  $U$  ranges from 0 to  $L - 1$  and  $V = 0$  to  $L - 1$ , we get Discrete image in frequency domain. Thus applying the 2D-DFT, we get the discrete version of the following matrix.

$$\text{MRIIMAGE}(x, y) = M_{xy}(x, y, 0^+) e^{\frac{-(T_{\text{readout}}/2)}{T_2(x,y)}} e^{j\phi} \quad (2.61)$$

$$\text{MRIDISCRETEIMAGE}(r, c) = \sum_0^{L-1} \sum_0^{L-1} C1(U, V) e^{\frac{j2\pi rU}{L}} e^{\frac{j2\pi cV}{L}} \quad (2.62)$$

Note that the  $r = 0$  and  $c = 0$  indicate that top-left corner position of the discrete image. Also note that  $r$  and  $c$  ranges from 0 to  $L - 1$ . Hence MRIDISCRETEIMAGE is obtained.

## 2.8 Practical Difficulties and Remedies in MRI

The Eqs.(2.47)–(2.50) completely describe the transverse components of the magnetic moment during the read-out duration. The equation is valid provided the slice under consideration must satisfy the following conditions.

- The  $\alpha$  value and the strong magnetic field  $B_0$  in the  $z$ -direction is constant along the  $z$ -axis. This is achived using RF exitation followed by refocussing gradient (as described in Sect. 2.5)
- There is no other sources that affect the magnetic field  $B_0$

But in practice, due to perturbation of the magnetic field  $B_0$  and other external magnetic field, the transverse components decreases at the faster rate with time constance  $T_2^*$  instead of  $T_2$ , with  $T_2^* \ll T_2$ . Hence the obtained discrete image described in the Sect. 2.7 is not completely due to  $T_2$  characteristics ( i.e spin–spin relaxation). If the signal  $s(G_x, G_y, t)$  is sampled at time instant  $t = 0$  (instead of  $T_{\text{readout}}/2$ ), the obtained image is proton-density MRI image.

### 2.8.1 Proton-Density MRI Image using Gradient Echo

In practice, sampling the signal  $s(G_x, G_y, t)$  at almost near to  $t = 0$  gives proton-density image and is achieved as follows.

1. The  $z$ -gradient and the RF pulse are applied simultaneously for the duration  $\tau_p/2$  to obtain the required  $\alpha$  value throughout the selected slice in the  $z$ -direction. (refer Sect. 2.5). The value of the  $\alpha$  is usually chosen as  $\frac{\pi}{2}$ .
2. Refocussing  $z$ -gradient is applied to obtain the identical magnetic field  $B_0$  in the  $z$ -axis, throughout the slice.
3. The transverse component at this moment is given as  $M_{xy}(x, y, t) = 2M_0 \sin(\alpha_{\tau_p/2}) e^{j(-\gamma(B_0)\tau_p + \phi)} e^{j(-\gamma(B_0)t)} e^{-\frac{t}{T_2(x,y)}}$
4. Apply the  $G_y$  gradient for the duration of  $\tau_y$ , so that the transverse component becomes  $M_{xy}(x, y, t) = 2M_0 \sin(\alpha_{\tau_p/2}) e^{j(-\gamma(B_0)\tau_p + \phi)} e^{j(-\gamma(B_0 + G_y y)\tau_y)} e^{j(-\gamma(B_0)t)} \times e^{-\frac{(t + \tau_y)}{T_2(x,y)}}$ .
5. Apply the  $-G_x$  gradient for the duration of  $\tau_x$ , so that the transverse component becomes  $M_{xy}(x, y, t) = 2M_0 \sin(\alpha_{\tau_p/2}) e^{j(-\gamma(B_0)\tau_p + \phi)} e^{j(-\gamma(B_0 + G_y y)\tau_y)} \times e^{j(-\gamma(B_0 - G_x x)\tau_x)} e^{j(-\gamma(B_0)t)} e^{-\frac{(t + \tau_y + \tau_x)}{T_2(x,y)}}$  (It is assumed that there is no significant change in  $\alpha$  value.)
6. The read-out phase starts at this moment. Postive gradient  $G_x$  is applied during the read-out phase for the duration of  $2\tau_x$ . The resultant magnetic moment during the read-out phase is given as mentioned in (2.63).

$$M_{xy}(x, y, t) = 2M_0 \sin(\alpha_{\tau_p/2}) e^{j(-\gamma(B_0)\tau_p + \phi)} e^{j(-\gamma(B_0 + G_y y)\tau_y)} e^{j(-\gamma(B_0 - G_x x)\tau_x)} \times e^{j(-\gamma(B_0 + G_x x)t)} e^{-\frac{(t + \tau_y + \tau_x)}{T_2(x,y)}}. \quad (2.63)$$

7. The phase component introduced due to  $G_x$  cancels in the middle of the read-out phase. This is known as Gradient echo. This helps to synchronize the hardware to sample the real and imaginary part of the signal at the end of the read-out phase ( $2\tau_x$ ) to obtain the sample value of the magnitude of the signal  $s(t)$  corresponding to the particular location in the K-space ((i.e)  $G_x, G_y$ ).
8. Wait for the complete relaxation ((i.e) equilibrium) until all the longitudinal components reaches  $B_0$ . This can also be done using spoiler gradient in modern techniques.

9. The steps 1–7 are repeated for the complete range of  $G_x$  and  $G_y$  (refer Sect. 2.7) and hence discrete MRI image in the frequency domain is obtained.
10. Apply the inverse 2D-DFT to obtain MRI image. The image thus obtained corresponds to proton-density image. This gives the proton-density (refer Sect. 3.1 for illustration) of every pixel of the image slice.

### 2.8.2 $T_2$ MRI Image Using Spin–Echo and Cartesian Scanning

$T_2$  MRI principles are explained with the micro-level behaviour of the randomly oriented individual magnetic moments (with various rate at which the phase is changing) at every point  $(x,y)$  across the slice. Due to the slice selection (along the  $z$ -axis) by applying the  $z$ -gradient, followed by RF excitation and refocussing gradient we are able to obtain the in-phase resultant magnetic moment making an angle  $\alpha$  with the  $z$ -axis (measured anti-clockwise when viewed along the  $z$ -axis). The corresponding transverse component is making an angle  $\phi$  with the  $x$ -axis (measured anti-clockwise when viewed along the  $z$ -axis).

The individual magnetic moments at the particular position  $(x, y)$  after the release of RF excitation, starts to experience different phase (even though it was made inphase due to external RF excitation). This is the natural phenomenon due to spin–spin interaction. The phase achieved by the individual magnetic moment over the time helps in decreasing the resultant transverse component. The rate at which the the phase of the individual magnetic moment is changing purely depends on the tissue. The rate at which the resultant transverse magnetic moment is decreasing is described by the factor  $T_2(x, y)$  and hence  $T_2(x, y)$  plays the important role in knowing the characteristics of the tissue and hence image is obtained. But in practice the rate at which the resultant transverse component is characterized by the factor  $T_2^*$ . Even after the transverse component becomes zero due to the factor  $T_2^*$ , the dephasing operation still continuous. This leads to the technique called spin echo (described below) to obtain the non-zero transverse component, (even after reaching zero due to  $T_2^*$ ). This in further helps to obtain the frequency sample of the MRI image highlighting the  $T_2$  values as described below.

1. Steps 1–4 are performed similar to the technique mentioned in the Sect. 2.8.1.
2. Thus the currently obtained transverse component is given by

$$M_{xy}(x, y, t) = 2M_0 \sin(\alpha\tau_p/2) e^{j(-\gamma(B_0)\tau_p + \phi)} e^{j(-\gamma(B_0 + G_y y)\tau_y)} e^{j(-\gamma(B_0 t))} e^{-\frac{(t + \tau_y)}{T_2(x, y)}}.$$

3. Apply the  $G_x$  gradient for the duration of  $\tau_x$ , so that the transverse component becomes (2.65).  $M_{xy}(x, y, t) = 2M_0 \sin(\alpha\tau_p/2) e^{j(-\gamma(B_0)\tau_p + \phi)} e^{j(-\gamma(B_0 + G_y y)\tau_y)} \dots$

$$e^{j(-\gamma(B_0 + G_x x)\tau_x)} e^{j(-\gamma(B_0 t))} e^{-\frac{(t + \tau_y + \tau_x)}{T_2(x, y)}} \quad (2.64)$$

It is assumed that there is no significant change in  $\alpha$  value.

4. The exponentially decreasing term  $e^{-\frac{t}{T_2(x,y)}}$  in (2.64) describes the micro-level behaviour of the individual magnetic moments (spin–spin interactions) at  $(x, y)$ . Thus the equation can also be written with micro-level behaviour of the individual magnetic moments as follows.

$$\sum_n M_{xy}(x, y, t, n) = \sum_n 2M_0 \sin(\alpha_{\tau_p}/2) e^{j(-\gamma(B_0)\tau_p + \phi_n(t,x,y))} e^{j(-\gamma(B_0+G_y y)\tau_y)} \\ \times e^{j(-\gamma(B_0+G_x x)\tau_x)} e^{j(-\gamma(B_0)t)}$$

where,  $M_{xy}(x, y, t, n)$  is the  $n$ th micro-level magnetic moment which is the function of  $x, y$  and  $t$ .

5. Now apply the  $180^\circ$  RF pulse. This is not same as that of the RF pulse. This helps in changing the phase component of the transverse component from arbitrary  $\rho$  to  $-\rho$ . Note that the selection gradient( $G_z$ ) is applied while applying  $180^\circ$  pulse.
6. After applying  $180^\circ$  pulse, the resultant transverse magnetic moment is given as

$$\sum_n M_{xy}(x, y, t, n) = \sum_n 2M_0 \sin(\alpha_{\tau_p}/2) e^{j(+\gamma(B_0)\tau_p - \phi_n(t,x,y))} e^{j(+\gamma(B_0+G_y y)\tau_y)} \\ \times e^{j(+\gamma(B_0+G_x x)\tau_x)} e^{j(-\gamma(B_0)t)}$$

7. The magnitude of the signal  $M_{xy}(x, y, t)$  at every pixel corresponding to the transverse component in step 5 is decreasing gradually with time (due to dephasing). Hence the magnitude of  $M_{xy}(x, y, t)$  corresponding to the transverse component in step 6 increases with time (refer Sect.3.2.2 for illustration) and reaches maximum after some time duration. This is known as spin–echo. Spin–echo guarantees the existence of required amplitude of MRI signal for sampling.
8. Read-out phase starts immediate after some time (required time for rephasing) after applying  $180^\circ$  pulse along with the positive  $x$ -gradient  $G_x$  for the duration of  $\tau_x$ . The resultant transverse component during read-out phase is given as

$$\sum_n M_{xy}(x, y, t, n) = \sum_n 2M_0 \sin(\alpha_{\tau_p}/2) e^{j(+\gamma(B_0)\tau_p - \phi_n(x,y,t))} e^{j(+\gamma(B_0+G_y y)\tau_y)} \\ \times e^{j(+\gamma(B_0+G_x x)\tau_x)} e^{j(-\gamma((B_0+G_x x)t))}$$

9. After time duration of  $\tau_x$ , there is the cancellation of the phase introduced due to  $G_x$  gradient (upto step 8). This is known as Gradient echo. This helps to synchronize the hardware and sample the magnitude of  $s(t)$  at the end of the reading phase  $2\tau_x$  corresponding to the particular location in the K-space. This step is same as that of the one used in proton-density imaging using phase

gradient. But what we achieved is the sampled value gives the information about  $T_2(x, y)$ . (refer Sect. 3.2) for illustration.

10. Note that the sampled value corresponds to  $(G_x, -G_y)$ , not  $(G_x, G_y)$ .

### 2.8.3 $T_2$ MRI Image Using Spin–Echo and Polar Scanning

1. Steps 1–3 are performed similar to the technique mentioned in the Sect. 2.8.1.
2. Thus the currently obtained transverse component is given by

$$M_{xy}(x, y, t) = 2M_0 \sin(\alpha\tau_p/2) e^{j(-\gamma(B_0)\tau_p + \phi)} e^{j(-\gamma(B_0)t)} e^{\frac{-t}{T_2(x,y)}}.$$

3. Apply both  $G_x$  and  $G_y$  gradient simultaneously for the time duration  $\tau_{xy}$ , so that the transverse components become the following.

$$M_{xy}(x, y, t) = 2M_0 \sin(\alpha\tau_p/2) e^{j(-\gamma(B_0)\tau_p + \phi)} e^{j(-\gamma(B_0 + G_y y + G_x x)\tau_{xy})} \\ \times e^{j(-\gamma(B_0)t)} e^{\frac{-(t + \tau_{xy})}{T_2(x,y)}}$$

4. The technique used in the steps 4–7 of the Sect. 2.8.2 are adopted.
5. Read-out phase starts immediate after some time (required time for rephasing) after applying  $180^\circ$  pulse along with the positive  $x$ -gradient  $G_x$  and positive  $y$ -gradient  $G_y$  for the duration of  $\tau_{xy}$ . The resultant transverse component during read-out phase is given as

$$\sum_n M_{xy}(x, y, t, n) = \sum_n 2M_0 \sin(\alpha\tau_p/2) e^{j(+\gamma(B_0)\tau_p - \phi_n(x,y,t))} \\ \times e^{j(+\gamma(B_0 + G_x x + G_y y)\tau_{xy})} e^{j(-\gamma((B_0 + G_x x + G_y y)t))}.$$

6. After time duration of  $\tau_{xy}$ , there is the cancellation of the phase introduced due to  $G_x$  and  $G_y$  gradient (upto step 4). This is gradient echo. This helps to synchronize the hardware to sample the magnitude of  $s(t)$  at the end of the reading phase  $2\tau_x$  corresponding to the particular location in the K-space. This step is same as that of the one used in phase gradient. But what we achieved is the sampled value gives the information about  $T_2(x, y)$ . (refer Sect. 3.3) for illustrations.
7. Note that the sample value corresponds to the point  $(-G_x, -G_y)$ .
8. This is the polar version of the technique used in Sect. 2.8.2. The main difference is that the gradients are applied simultaneously. Also the value of  $G_x$  and  $G_y$  are changed in such a way that the samples of the k-space (frequency domain) are uniformly scanned over the variables  $r$  and  $\theta$ , where  $r = \sqrt{(G_x^2 + G_y^2)}$  and  $\theta = \tan^{-1} \left( \frac{G_y}{G_x} \right)$ .

### 2.8.4 $T_1$ MRI Image

We understand that there is the natural relaxation in the transverse component of the resultant magnetic moment due to spin–spin interaction. This is known as  $T_2$  relaxation. We exploit the different relaxation time and different proton-density (refer Sects. 2.8.1–2.8.3) for the different tissues to obtain MRI image ( $T_2$  and proton-density respectively). In the same way, due to spin–lattice interaction, the longitudinal components gradually increases. This is the another natural relaxation known as  $T_1$  relaxation (which is independent of  $T_2$  relaxation). Different tissues have different  $T_1$  relaxation time. This helps to obtain the another type of MRI image known as  $T_1$  MRI image. Usually  $T_1$  is very larger when compared with the corresponding  $T_2$ . Exploiting this property, the following trick is used to obtain the  $T_1$  MRI image.

1. Steps 1–3 in Sect. 2.8.1 is performed.
2. Wait for the time to get almost zero transverse-component. But by this time, longitudinal component gradually increases (not reached maximum).
3. At this moment, flip the longitudinal component by an angle  $\alpha$  (usually  $90^\circ$ ) to obtain the transverse component. Sample the transverse component immediately after obtaining the transverse component (i.e at  $t = 0$  during read-out phase) to obtain the K-space of the  $T_1$  image for the particular  $G_x$  and  $G_y$ . The operation is repeated for the required range of  $G_x$  and  $G_y$  values. Apply 2D-IDFT to obtain the  $T_1$  image. Instead of sampling the transverse component at time instant  $t = 0$ , gradient echo can also be used to measure the transverse component as described in the Sect. 2.8.1.
4. Note that  $T_1$  image is obtained by measuring the transverse component. But the measured transverse component is mainly due to longitudinal relaxation obtained from spin–lattice interaction. The trick is to flip the longitudinal component, which are in relaxation state to obtain the transverse component. The magnitude of the transverse component describes the  $T_1$  characteristics of the tissue. Thus the image obtained is known as  $T_1$  image (refer Sect. 3.4 for illustration).

# Chapter 3

## Illustrations on MRI Techniques

### Using Matlab

#### 3.1 Illustration on the Steps Involved in Obtaining Proton-Density MRI Image

Let the image slice in the z-direction is selected and  $90^\circ$  RF pulse is applied. Refocussing gradient is also applied to compensate the phase introduced during the slice selection. Let the size of the image slice be  $11 \times 11$  in spatial domain. Under this condition, every pixel of the image slice  $(x, y)$  are assumed to have the finite number of identical individual transverse magnetic moments  $n(x, y)$ . Also the individual magnetic moment at every position is undergoing linear phase delay with different rate  $k(x, y, i)$  i.e.  $\phi(t) = 2\pi k(x, y, i)t$ . Thus every pixel of the image slice is characterized by the following factors.

- $n(x, y)$ : Number of individual magnetic moments corresponding to proton density.
- $k(x, y, i)$ : Phase constant associated with  $i$ th magnetic moment of the  $(x, y)$  position of the image slice.

When the individual magnetic moments of the particular position  $(x, y)$  are undergoing random phase change, resultant magnitude of the magnetic moment at  $(x, y)$  gradually decreases at every pixel. This is known as dephasing. The rate at which the magnitude is decreasing gives the characteristics of the image slice at  $(x, y)$ . In case of proton-density image, we try to record  $n(x, y)$  and in the case of  $T_2$  image, we try to measure the magnitude of the resultant magnetic moment of every pixel  $(x, y)$  at the particular instant to sense the rate at which the magnitude is decreasing i.e.  $T_2$  imaging. The time instants mentioned in this chapter are arbitrarily chosen and are represented in seconds.



### 3.1.1 Proton-Density MRI Imaging

Let the image slice be represented as the matrix with the values filled up with proton-density is as follows.

$$n = \begin{bmatrix} 3 & 3 & 9 & 7 & 9 & 5 & 8 & 6 & 9 & 9 & 4 \\ 5 & 5 & 9 & 10 & 9 & 5 & 5 & 1 & 7 & 9 & 10 \\ 6 & 9 & 6 & 1 & 2 & 9 & 9 & 2 & 6 & 9 & 6 \\ 5 & 6 & 2 & 2 & 7 & 8 & 2 & 2 & 2 & 9 & 2 \\ 9 & 8 & 1 & 10 & 9 & 5 & 8 & 7 & 10 & 1 & 5 \\ 6 & 1 & 2 & 4 & 7 & 3 & 7 & 8 & 1 & 3 & 9 \\ 9 & 9 & 7 & 6 & 5 & 4 & 7 & 9 & 10 & 3 & 4 \\ 5 & 3 & 2 & 9 & 8 & 9 & 4 & 6 & 5 & 10 & 7 \\ 4 & 1 & 3 & 8 & 2 & 9 & 5 & 10 & 9 & 4 & 6 \\ 5 & 9 & 6 & 4 & 4 & 2 & 1 & 7 & 8 & 9 & 7 \\ 4 & 8 & 5 & 7 & 8 & 3 & 1 & 4 & 9 & 8 & 4 \end{bmatrix}$$

These are the magnitude of the resultant magnetic moment at  $t = 0$  (refer Fig. 3.4). The following steps are followed to obtain the K-space corresponding to the proton-density image. The identical transverse components are assumed to be available at every pixel of the image slice, i.e. they are in-phase and are rotating in the Larmor frequency.

1. Apply  $G_y$  gradient for the duration of 0.0000000009.
2. Apply  $-G_x$  gradient for the duration of 0.0000000009.
3. Apply  $G_x$  gradient for the duration of 0.0000000018. During this phase, after the time duration of 0.0000000009, there is the cancellation of phase introduced due to  $G_x$ . This is known as Gradient echo. This helps to synchronize the hardware to sample at the particular instant during real time to choose the proper position in the K-space.
4. Sample the real and imaginary component of the signal  $s(t)$  to obtain the sample of the K-space at  $(G_x, G_y)$ .
5. Proper scaling factor is used so that the final basis (to perform IFFT2) look like the standard form  $e^{-\frac{j2\pi x G_x}{\Pi}} e^{-\frac{j2\pi y G_y}{\Pi}}$ . This is the process of discretization.
6. The above steps are repeated for the complete scan in the K-space. For every time, we have to wait for the longitudinal component to reach maximum before applying  $90^\circ$  RF pulse. (This is illustrated by repeating the steps 1–5 by varying the values of  $G_x$  and  $G_y$ , ranging from  $-5$  to  $5$  with the interval of 1.)

The resultant KSPACE is obtained as follows.

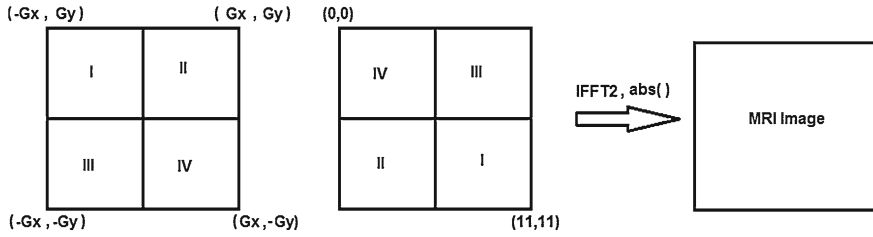


Fig. 3.1 Original and reconstructed K-space for getting MRI image

$$\begin{aligned}
 \text{Re}(KSPACE) &= \begin{bmatrix} -4.71 & -12.82 & -2.89 & 57.96 & -5.48 & 11.31 & 3.35 & -39.67 & -24.11 & -0.34 & 19.19 \\ 26.47 & -9.98 & -24.29 & -4.29 & -28.18 & -23.23 & 0.67 & 10.63 & 3.04 & 1.45 & 12.73 \\ 7.42 & 31.81 & 59.29 & -15.34 & -33.04 & 15.16 & -46.07 & -16.54 & -22.91 & -2.44 & -11.88 \\ -4.69 & 6.59 & 12.34 & -0.29 & -1.74 & 30.58 & -8.22 & -30.34 & 4.82 & -12.82 & 18.14 \\ 23.15 & -18.67 & -49.30 & -24.40 & 2.44 & 11.05 & 34.95 & -37.86 & -37.53 & 12.97 & -11.28 \\ -9.51 & 7.10 & 18.70 & -30.82 & -6.02 & 693.89 & 2.54 & -20.17 & 10.91 & 1.68 & -10.68 \\ -18.21 & 12.50 & -23.29 & -56.08 & 36.61 & 13.42 & -0.81 & -33.28 & -51.17 & -20.94 & 26.57 \\ 3.23 & -17.01 & 9.72 & -29.82 & -0.19 & 20.29 & -4.93 & -4.56 & 35.67 & 0.20 & 5.92 \\ -3.89 & -2.59 & -28.57 & -26.29 & -37.39 & 10.46 & -33.10 & -5.57 & 55.70 & 28.03 & 10.00 \\ 13.84 & -3.24 & -3.26 & 24.46 & -12.80 & -31.76 & -31.32 & -3.83 & -26.37 & -4.38 & 25.49 \\ 7.97 & 13.58 & -16.73 & -32.80 & -5.08 & 25.05 & -3.15 & 50.89 & 2.87 & -3.73 & -10.30 \end{bmatrix} \\
 \text{Im}(KSPACE) &= \begin{bmatrix} 15.49 & -20.75 & -14.20 & 6.40 & -4.86 & -37.58 & 20.98 & -9.61 & -14.07 & -35.68 & 24.91 \\ -2.84 & -12.35 & 10.26 & -0.30 & 13.79 & 26.61 & 34.46 & -37.66 & 15.55 & 11.75 & -5.44 \\ -8.14 & 3.26 & -2.80 & -21.96 & 6.83 & 8.99 & -12.94 & 28.40 & 19.17 & 0.87 & -18.12 \\ -26.31 & 15.07 & -62.43 & 11.02 & 8.55 & 20.22 & -18.95 & 4.80 & -13.54 & 13.36 & 34.59 \\ -13.48 & 9.60 & 14.80 & 27.75 & 7.85 & -8.34 & -11.35 & 54.47 & -28.97 & -1.42 & 20.08 \\ 4.94 & 12.48 & 16.21 & -21.11 & -20.77 & -140.66 & 21.48 & 31.45 & -22.21 & -14.26 & -0.85 \\ -14.09 & -3.74 & 41.30 & -35.42 & -3.16 & 3.38 & -8.18 & -16.06 & 5.57 & -1.57 & 3.40 \\ -38.92 & -7.31 & 10.60 & 7.40 & 20.65 & -30.53 & -7.20 & -10.03 & 52.69 & -16.45 & 26.06 \\ 21.31 & 0.15 & -8.73 & -19.72 & 29.85 & -14.18 & 6.57 & 26.20 & -20.51 & -15.39 & 4.61 \\ 0.06 & -11.38 & -15.50 & 30.55 & -32.00 & -15.46 & -1.73 & 1.94 & 0.01 & 15.27 & -7.68 \\ -30.42 & 33.00 & 22.35 & 24.30 & -20.63 & 30.20 & 6.61 & -28.47 & 14.20 & 24.10 & -12.44 \end{bmatrix}
 \end{aligned}$$

(1, 1) position of the matrix  $Re(KSPACE)$  corresponds to the frequency  $(-G_x, -G_y)$ . Similarly (11, 11) corresponds to  $(G_x, -G_y)$ . Hence K-space is rearranged properly (refer Fig. 3.1) before applying IFFT2.

IFFT2 is applied to the reconstructed K-space (refer Fig. 3.1) and the absolute value is computed to obtain the image matrix as shown below.

$$\text{FINALPROTONDENSITYIMAGE} = \begin{bmatrix} 3 & 3 & 9 & 7 & 9 & 5 & 8 & 6 & 9 & 9 & 4 \\ 5 & 5 & 9 & 10 & 9 & 5 & 5 & 1 & 7 & 9 & 10 \\ 6 & 9 & 6 & 1 & 2 & 9 & 9 & 2 & 6 & 9 & 6 \\ 5 & 6 & 2 & 2 & 7 & 8 & 2 & 2 & 2 & 9 & 2 \\ 9 & 8 & 1 & 10 & 9 & 5 & 8 & 7 & 10 & 1 & 5 \\ 6 & 1 & 2 & 4 & 7 & 3 & 7 & 8 & 1 & 3 & 9 \\ 9 & 9 & 7 & 6 & 5 & 4 & 7 & 9 & 10 & 3 & 4 \\ 5 & 3 & 2 & 9 & 8 & 9 & 4 & 6 & 5 & 10 & 7 \\ 4 & 1 & 3 & 8 & 2 & 9 & 5 & 10 & 9 & 4 & 6 \\ 5 & 9 & 6 & 4 & 4 & 2 & 1 & 7 & 8 & 9 & 7 \\ 4 & 8 & 5 & 7 & 8 & 3 & 1 & 4 & 9 & 8 & 4 \end{bmatrix}$$

In this procedure, it is noted that at the time of sampling the K-space, there is no significant dephasing experienced by the individual magnetic moments at every pixel. Hence the reconstructed image mainly contributes the proton-density at every pixel. It is observed from the matrix FINALPROTONDENSITYIMAGE that the pixel values are exactly the number of individual magnetic moments ( $n$ ) at every pixel ( $x, y$ ). This type of MRI imaging technique is known as proton-density MRI imaging.

### 3.1.1.1 protondensity.m

```

siz=11;
K=0.2; %Initial phase
Mxyit=[];
gamma=2*pi*42.58*(10^6);
d1=((10^3)/(11*42.58*(10^6)*0.000000009));
d2=((10^3)/(11*42.58*(10^6)*0.000000009));
B0=1;
n=round(rand(siz,siz)*9+1);
for x=1:1:siz
    for y=1:1:siz
        for i=1:1:n(x,y)
            k(x,y,i)=rand*2-1;
        end
    end
end
KSPACE=zeros(11,11);
for Gx=-5:1:5;
for Gy=-5:1:5;
Gy1=Gy/(10^3);%This is to indicate that B0 is high compared with y-gradient
Gx1=Gx/(10^3);%This is to indicate that B0 is high compared with x-gradient
for x=0:1:siz-1
for y=0:1:siz-1
for i=1:1:n(x+1,y+1)
phi1=k(x+1,y+1,i)*0.000000009*2*pi;
%The magnetic moments after applying y-gradient for the duration of
%0.000000009
temp1(x+1,y+1,i)=exp(-j*((B0*gamma+gamma*y*d2*Gy1)*0.000000009-phi1+K));
%The magnetic moments after applying x-gradient for the duration of
%0.000000009 after applying the y-gradient for the duration of
%0.000000009
phi2=k(x+1,y+1,i)*0.000000009*2*pi;
temp2(x+1,y+1,i)=exp(-j*((B0*gamma+d2*gamma*y*Gy1)*0.000000009-phi1+K))...
*exp(-j*((B0*gamma-gamma*d1*x*Gx1)*0.000000009-phi2));
%The magnetic moments after applying x-gradient for the duration of
%0.0000000018 after applying the y-gradient for the duration of
%0.000000009 and negative x-gradient for the duration of 0.000000009
phi3=k(x+1,y+1,i)*0.000000018*2*pi;
temp3(x+1,y+1,i)=exp(-j*((B0*gamma+gamma*d2*y*Gy1)*0.000000009-phi1+K))...
*exp(-j*((B0*gamma-gamma*d1*x*Gx1)*0.000000009-phi2))...
exp(-j*((B0*gamma+gamma*d1*x*Gx1)*0.000000018-phi3));
end
end
end

for x=0:1:siz-1
    for y=0:1:siz-1
        for i=1:1:n(x+1,y+1)

```

```

        KSPACE(Gx+6,Gy+6)=KSPACE(Gx+6,Gy+6)+temp3(x+1,y+1,i);
    end
end
end
end

KSPACEREARRANGED=KSPACE(6:1:11,6:1:11);
KSPACEREARRANGED=[KSPACEREARRANGED KSPACE(6:1:11,1:1:5)];
KSPACEREARRANGED=[KSPACEREARRANGED; KSPACE(1:1:5,6:1:11) KSPACE(1:1:5,1:1:5) ];
RECONSTRUCTED=ifft2(KSPACEREARRANGED)
%Hence the measured image is the proton density image.

```

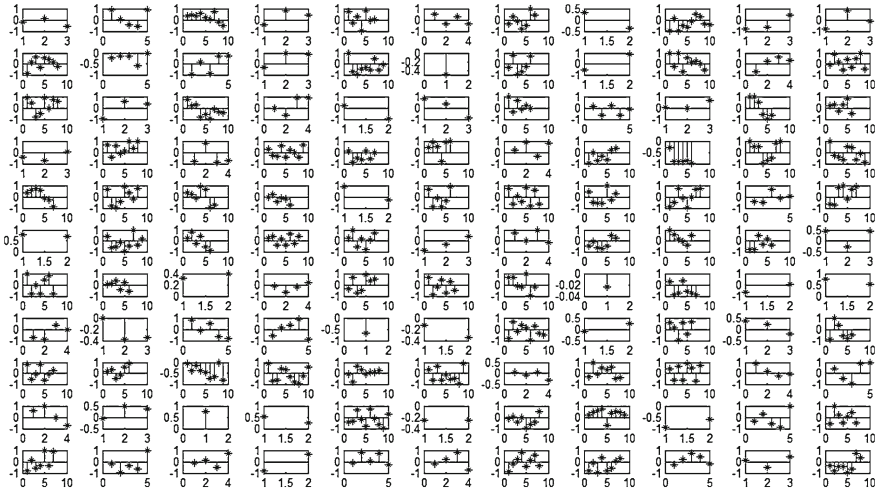
### 3.2 Illustration on the Steps Involved in Obtaining the $T_2$ MRI Image Using Cartesian Scanning

Let the image slice be represented as the matrix with the values filled up with proton-density (number of individual magnetic moments available at every pixel) is as follows.

$$n = \begin{bmatrix} 3 & 5 & 9 & 3 & 7 & 4 & 7 & 2 & 10 & 3 & 3 \\ 8 & 5 & 5 & 3 & 9 & 1 & 6 & 2 & 9 & 4 & 9 \\ 8 & 3 & 9 & 4 & 2 & 3 & 6 & 5 & 3 & 6 & 6 \\ 3 & 8 & 4 & 9 & 7 & 6 & 4 & 7 & 6 & 8 & 9 \\ 7 & 8 & 7 & 6 & 2 & 6 & 8 & 7 & 8 & 5 & 7 \\ 2 & 9 & 6 & 8 & 7 & 3 & 4 & 7 & 6 & 6 & 3 \\ 7 & 6 & 2 & 4 & 7 & 7 & 7 & 1 & 7 & 2 & 2 \\ 4 & 3 & 5 & 5 & 1 & 2 & 9 & 2 & 6 & 3 & 6 \\ 7 & 6 & 9 & 10 & 8 & 9 & 4 & 8 & 8 & 4 & 5 \\ 4 & 3 & 1 & 2 & 10 & 2 & 8 & 9 & 2 & 5 & 7 \\ 7 & 5 & 4 & 2 & 5 & 4 & 8 & 8 & 5 & 3 & 8 \end{bmatrix}$$

Every magnetic moments at every pixel are assumed to be identical (magnitude and phase) at  $t = 0$ . Initial phase of every individual magnetic moments are assumed to be identically equal to  $K = 0.2$ . Individual magnetic moments of every pixel undergoes phase change, which varies linearly as  $2\pi k(x, y, i)t$ , where  $k(x, y, i)$  is the linear phase constant associated with  $i$ th magnetic moment at the position  $(x, y)$ . Phase constants of the individual magnetic moments at every pixel are illustrated in the Fig. 3.2. Every subplot of the Fig. 3.2 belongs to the individual pixel. For instant (1, 1) subplot corresponds to (1, 1) pixel. The phase constants associated with the three (note that  $n(1, 1) = 3$ ) individual magnetic moments at the position (1, 1) is shown as the three stems in the subplot (1, 1). The following steps are followed to obtain  $T_2$  image.

1. Apply  $G_y$  gradient for the duration of 0.1.
2. Apply  $G_x$  gradient for the duration of 0.1.



**Fig. 3.2** Illustration of the phase constants of the individual magnetic moments at every pixel of the image slice of size  $11 \times 11$  (refer Sect. 3.2)

3. Apply  $180^\circ$  pulse and allow the magnetic moment to dephase for 0.3. During this phase, after the time duration of 0.2, there exists the momentary peak occur in the resultant magnitude of the magnetic moment of every pixel. This is called spin echo (refer Sect. 3.2.2). Usually the transverse components reduces drastically due to factors other than the spin-spin relaxation. The transverse component becomes almost zero before the hardware perform the procedure to capture the MRI signal. But spin echo guarantees the availability of the signal, so that the hardware can capture the required signal.
4. Apply  $G_x$  gradient for the duration of 0.2. During this period, there is the cancellation of phase introduced due to the  $G_x$  component. This is known as Gradient echo. This helps to synchronize the hardware to sample at the particular sample time to choose the proper position in the K-space. This is similar to the one used in proton-density MRI image.
5. Sample the real and imaginary component of the signal  $s(t)$  to obtain the sample of the K-space at  $(G_x, -G_y)$ .
6. Proper scaling factor is used so that the final basis (to perform IFFT2) look like the standard form  $e^{-\frac{j2\pi x G_x}{11}} e^{-\frac{j2\pi y G_y}{11}}$ . This is the process of discretization.
7. The above steps are repeated for the complete scan in the K-space. For every time, we have to wait for the longitudinal component to reach maximum before applying  $90^\circ$  RF pulse. (This is illustrated by repeating the steps 1–5 by varying the values of  $G_x$  and  $G_y$  ranging from  $-5$  to  $5$  with the interval of 1).

The obtained K-space is viewed as the function of  $G_x$  and  $G'_y = -G_y$ . The K-space is rearranged and are subjected to IFFT2 to obtain the MRI image  $f(x, -y)$

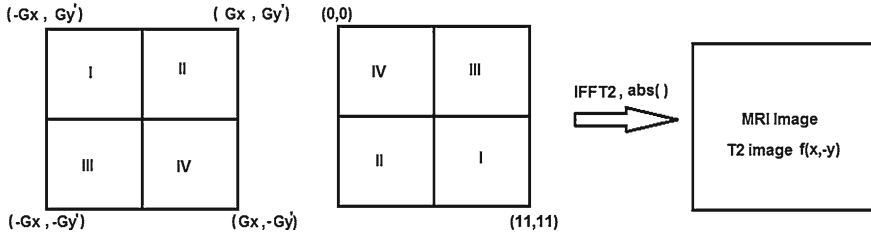


Fig. 3.3 Original and reconstructed K-space for getting MRI  $T_2$  image  $f(x, -y)$

(refer Fig. 3.3). The image in the spatial domain  $f(x, -y)$  is further properly arranged to obtain  $f(x, y)$ .

Thus the reconstructed  $T_2$  image matrix is as follows.

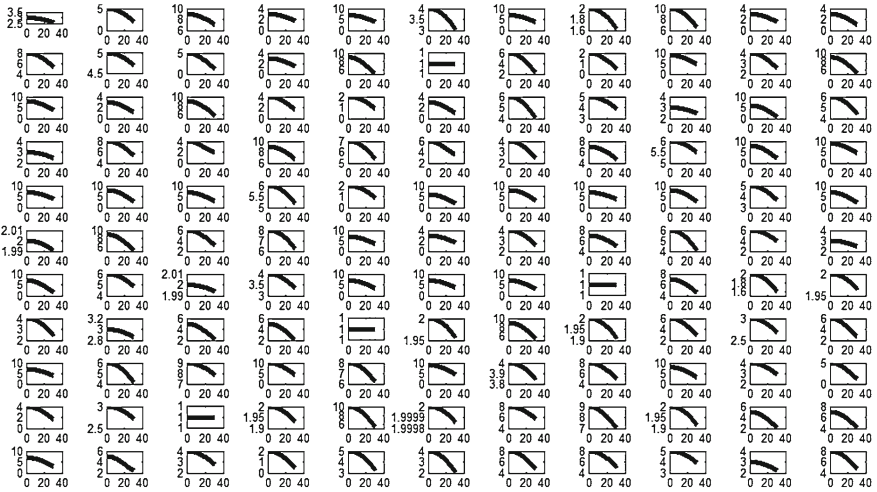
$$\text{RECONSTRUCTEDIMAGE} = \begin{bmatrix} 2.55 & 2.00 & 6.99 & 1.73 & 3.92 & 3.04 & 3.76 & 1.62 & 6.54 & 1.55 & 1.02 \\ 5.26 & 4.70 & 1.08 & 1.47 & 5.04 & 1.00 & 2.22 & 0.42 & 4.63 & 2.53 & 5.02 \\ 3.89 & 1.17 & 5.57 & 1.60 & 0.88 & 1.01 & 4.02 & 3.85 & 2.48 & 0.74 & 4.39 \\ 2.44 & 5.30 & 2.00 & 6.65 & 5.32 & 3.64 & 2.41 & 4.64 & 5.49 & 2.30 & 4.54 \\ 4.00 & 2.67 & 2.91 & 5.19 & 0.88 & 1.97 & 3.18 & 3.78 & 2.74 & 3.65 & 2.18 \\ 2.00 & 5.19 & 3.01 & 6.21 & 3.47 & 1.61 & 2.52 & 4.92 & 4.08 & 3.92 & 2.43 \\ 1.49 & 4.83 & 2.00 & 3.33 & 3.30 & 3.66 & 3.13 & 1.00 & 4.52 & 1.59 & 1.96 \\ 2.40 & 2.85 & 2.01 & 2.00 & 1.00 & 1.96 & 5.30 & 1.91 & 2.90 & 2.67 & 2.66 \\ 3.87 & 4.12 & 7.80 & 4.25 & 6.18 & 4.52 & 3.84 & 4.84 & 3.05 & 2.89 & 1.13 \\ 1.60 & 2.71 & 1.00 & 1.93 & 5.29 & 2.00 & 5.75 & 7.01 & 1.91 & 2.13 & 4.02 \\ 2.96 & 2.41 & 2.73 & 0.42 & 3.23 & 2.07 & 4.69 & 4.96 & 3.90 & 2.22 & 4.76 \end{bmatrix}$$

The resultant magnitude of the magnetic moments at every pixel during dephasing (refer Figs. 3.4, 3.5). Note that at  $t = 0$ , the value is the corresponding proton-density. As time goes, the resultant magnitude gradually decreases. At  $t = 0.3$ , (note that we allowed the magnetic moment to dephase for 0.3) the resultant magnitude (refer Fig. 3.4) is measured as follows.

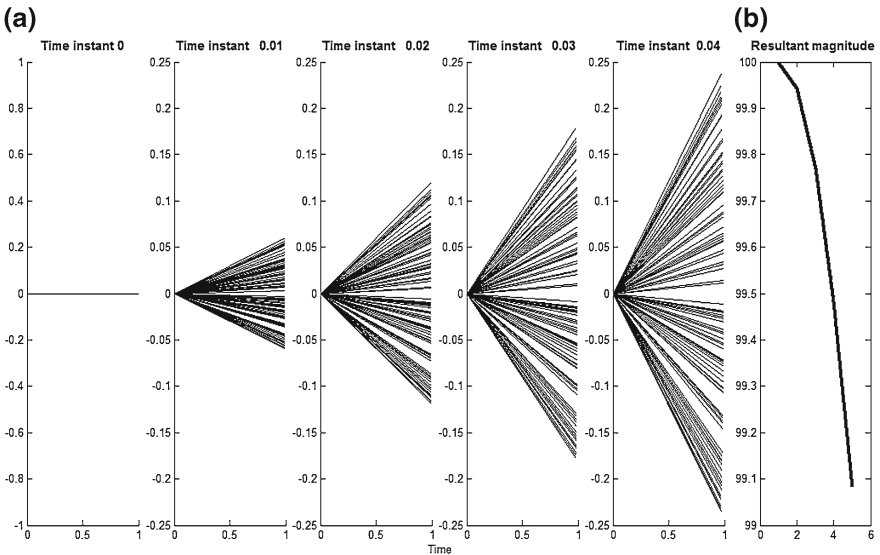
$$\text{RESULTANTMAGNITUDE}_{at t=0.3} = \begin{bmatrix} 2.55 & 2.00 & 6.99 & 1.73 & 3.92 & 3.04 & 3.76 & 1.62 & 6.54 & 1.55 & 1.02 \\ 5.26 & 4.70 & 1.08 & 1.47 & 5.04 & 1.00 & 2.22 & 0.42 & 4.63 & 2.53 & 5.02 \\ 3.89 & 1.17 & 5.57 & 1.60 & 0.88 & 1.01 & 4.02 & 3.85 & 2.48 & 0.74 & 4.39 \\ 2.44 & 5.30 & 2.00 & 6.65 & 5.32 & 3.64 & 2.41 & 4.64 & 5.49 & 2.30 & 4.54 \\ 4.00 & 2.67 & 2.91 & 5.19 & 0.88 & 1.97 & 3.18 & 3.78 & 2.74 & 3.65 & 2.18 \\ 2.00 & 5.19 & 3.01 & 6.21 & 3.47 & 1.61 & 2.52 & 4.92 & 4.08 & 3.92 & 2.43 \\ 1.49 & 4.83 & 2.00 & 3.33 & 3.30 & 3.66 & 3.13 & 1.00 & 4.52 & 1.59 & 1.96 \\ 2.40 & 2.85 & 2.01 & 2.00 & 1.00 & 1.96 & 5.30 & 1.91 & 2.90 & 2.67 & 2.66 \\ 3.87 & 4.12 & 7.80 & 4.25 & 6.18 & 4.52 & 3.84 & 4.84 & 3.05 & 2.89 & 1.13 \\ 1.60 & 2.71 & 1.00 & 1.93 & 5.29 & 2.00 & 5.75 & 7.01 & 1.91 & 2.13 & 4.02 \\ 2.96 & 2.41 & 2.73 & 0.42 & 3.23 & 2.07 & 4.69 & 4.96 & 3.90 & 2.22 & 4.76 \end{bmatrix}$$

(which is same as that of RECONSTRUCTEDIMAGE).

Thus  $T_2$  image gives the resultant magnitude of the magnetic moment at every pixel after dephasing for some finite time duration (0.3 in our illustration). We also know that the initial values of the magnetic moments ( $t = 0$ ) at every pixel is



**Fig. 3.4** Illustration of the decrease in the magnitude of the resultant magnetic moment at every pixel (of the  $11 \times 11$  image slice) due to natural transverse relaxation (spin-spin interaction based)



**Fig. 3.5** **a** Illustration of the dephasing of the individual magnetic moments at one particular pixel position ( $x, y$ ); **b** corresponding illustration of the decrease in the magnitude of the resultant magnetic moment

measured using proton-density image. Let the resultant magnitude of the magnetic moment of the pixel  $(x, y)$  at  $t = 0$  and at  $t = 0.3$  be represented as  $K(x, y)_0$  and  $K(x, y)_{0.3}$  respectively. They are related as  $K(x, y)_{0.3} = K(x, y)_0 e^{\frac{-0.3}{T_2(x, y)}}$ . Hence

$$T_2(x, y) = 0.3 * \log_e \left( \frac{K(x, y)_{0.3}}{K(x, y)_0} \right) \quad (3.1)$$

Thus  $T_2$  image is obtained as follows.

$$\text{FINAL}T_2\text{IMAGE} = \begin{bmatrix} 0.16 & 0.92 & 0.2527 & 0.5505 & 0.5798 & 0.2744 & 0.6215 & 0.2107 & 0.4246 & 0.6604 & 1.0788 \\ 0.42 & 0.06 & 1.5325 & 0.7133 & 0.5798 & 0 & 0.9943 & 1.5606 & 0.6647 & 0.4581 & 0.5838 \\ 0.72 & 0.94 & 0.4798 & 0.9163 & 0.8210 & 1.0887 & 0.4005 & 0.2614 & 0.1904 & 2.0929 & 0.3124 \\ 0.21 & 0.41 & 0.6931 & 0.3026 & 0.2744 & 0.4998 & 0.5067 & 0.4112 & 0.0888 & 1.2465 & 0.6843 \\ 0.56 & 1.0974 & 0.8778 & 0.1450 & 0.8210 & 1.1137 & 0.9226 & 0.6162 & 1.0715 & 0.3147 & 1.1666 \\ 0 & 0.5505 & 0.6898 & 0.2533 & 0.7018 & 0.6224 & 0.4620 & 0.3526 & 0.3857 & 0.4257 & 0.2107 \\ 1.55 & 0.2169 & 0 & 0.1833 & 0.7520 & 0.6484 & 0.8049 & 0 & 0.4374 & 0.2294 & 0.0202 \\ 0.51 & 0.0513 & 0.9113 & 0.9163 & 0 & 0.0202 & 0.5295 & 0.0460 & 0.7270 & 0.1165 & 0.8134 \\ 0.59 & 0.3759 & 0.1431 & 0.8557 & 0.2581 & 0.6887 & 0.0408 & 0.5025 & 0.9643 & 0.3250 & 1.4872 \\ 0.92 & 0.1017 & 0 & 0.0356 & 0.6368 & 0 & 0.3302 & 0.2499 & 0.0460 & 0.8533 & 0.5546 \\ 0.86 & 0.7298 & 0.3820 & 1.5606 & 0.4370 & 0.6587 & 0.5340 & 0.4780 & 0.2485 & 0.3011 & 0.5192 \end{bmatrix}$$

### 3.2.1 Note to the Fig. 3.4

1. The initial magnitude of the resultant magnetic moment gives the proton-density of the particular pixel. This leads to proton-density image.
2. The rate at which the resultant magnetic moment is decreasing gives the time constant  $T_2$ , which leads to  $T_2$  image.

#### 3.2.1.1 t2.m

```
siz=11;
K=0.2; %Initial phase
Mxyit=[];
gamma=2*pi*42.58*(10^6);
d1=((10^3)/(11*42.58*10^6*0.1));
d2=((10^3)/(11*42.58*10^6*0.1));
B0=1;
n=round(rand(siz,siz)*9+1);
%n=ones(siz,siz)*10;
for x=1:1:siz
    for y=1:1:siz
        for i=1:1:n(x,y)
            k(x,y,i)=rand*2-1;
        end
    end
end
KSPACE=zeros(11,11);
for Gx=-5:1:5;
for Gy=-5:1:5;
```



```

Gx1=Gx/(10^3);
Gy1=Gy/(10^3);
for x=0:1:siz-1
for y=0:1:siz-1
for i=1:1:n(x+1,y+1)
phi1=k(x+1,y+1,i)*0.1*2*pi;
%The magnetic moments after applying positive y-gradient for the duration of
%0.1
temp1(x+1,y+1,i)=exp(-j*((B0*gamma+gamma*y*d2*Gy1)*0.1-phi1+K));
%The magnetic moments after applying negative x-gradient for the duration of
%0.1 after applying the y-gradient for the duration of 0.1

phi2=k(x+1,y+1,i)*0.1*2*pi;
temp2(x+1,y+1,i)=exp(-j*((B0*gamma+d2*gamma*y*Gy1)*0.1-phi1+K))...
*exp(-j*((B0*gamma+gamma*d1*x*Gx1)*0.1-phi2));
%Apply 180 degree phase shift and allow it to dephase for 0.3 duration
phi3=k(x+1,y+1,i)*0.3*2*pi;
temp3(x+1,y+1,i)=exp(j*((B0*gamma+gamma*d2*y*Gy1)*0.1-phi1+K))...
*exp(j*((B0*gamma+gamma*d1*x*Gx1)*0.1-phi2))*...
exp(-j*((B0*gamma)*0.3-phi3));

%apply positive x-gradient for readout for the duration of 0.2

phi4=k(x+1,y+1,i)*0.2*2*pi;
temp3(x+1,y+1,i)=exp(j*((B0*gamma+gamma*d2*y*Gy1)*0.1-phi1+K))...
*exp(j*((B0*gamma+gamma*d1*x*Gx1)*0.1-phi2))*...
exp(-j*((B0*gamma)*0.3-phi3))*exp(-j*((B0*gamma+gamma*d1*x*Gx1)*0.2-phi4));

end
end
end

for x=0:1:siz-1
for y=0:1:siz-1
for i=1:1:n(x+1,y+1)
KSPACE(Gx+6,Gy+6)=KSPACE(Gx+6,Gy+6)+temp3(x+1,y+1,i);
end
end
end

KSPACEREARRANGED=KSPACE(6:1:11,6:1:11);
KSPACEREARRANGED=[KSPACEREARRANGED KSPACE(6:1:11,1:1:5)];
KSPACEREARRANGED=[KSPACEREARRANGED; KSPACE(1:1:5,6:1:11) KSPACE(1:1:5,1:1:5) ];
RECONSTRUCTED=abs(fft2(KSPACEREARRANGED));
%But note that what we reconstructed is f(x,-y) and not f(x,y).
save T2DATA n k
FINALRECONSIMAGE=zeros(11,11);
FINALRECONSIMAGE(:,1)=RECONSTRUCTED(:,1);
for p=1:1:11
for q=2:1:11
FINALRECONSIMAGE(p,q)=RECONSTRUCTED(p,13-q);
end
end
end

```

### 3.2.1.2 Relaxation.m

```

load T2DATA
vector=0
for x=1:1:11
    for y=1:1:11
        MAG{x,y}=[];
        for t=0:0.01:0.3
            vector=0;
            for i=1:1:n(x,y)
                phi=k(x,y,i)*t*2*pi;
                vector=vector+exp(-j*phi);
            end
            MAG{x,y}=[MAG{x,y} abs(vector)];
        end
    end
end
end
figure
sub=1;
for x=1:1:11
    for y=1:1:11
        subplot(11,11,sub)
        %plot(MAG{x,y}/max(MAG{x,y}))
        plot(MAG{x,y})
        sub=sub+1;
        COLLECTION(x,y)=MAG{x,y}(length(MAG{x,y}));
    end
end
end

```

### 3.2.1.3 Dephasing.m

```

%Concept of dephasing
MAG=[];
t=0:0.01:1;
vectorsum=0;
k=rand(1,100)*2-1;
snap=1;
for t=0:0.01:0.04
    vectorsum=0;
    for i=1:1:100
        phi=k(i)*t*2*pi;
        subplot(1,6,snap)
        hold on
        plot([0 cos(phi)],[0 -sin(phi)], 'k')
        title(strcat('Time instant',num2str(t)))
        vectorsum=vectorsum+exp(-j*phi);
    end
    MAG=[MAG vectorsum];
    snap=snap+1;
end
subplot(1,6,snap)
plot(abs(MAG), 'k')
title('Resultant magnitude versus time')
xlabel('Time')

```

### 3.2.2 Momentary Peak Due to Spin Echo

The following experiment demonstrate the momentary peak of the resultant magnitude at every pixel achieved after applying  $180^\circ$  pulse. This is due to rephasing. This is called spin echo. Spin echo helps to obtain the significant signal for a while so that the hardware can capture the required signal during dephasing. The experiment involves obtaining the spatial domain image at three different time instances as described below.

Time instant: 0.2

1. Initialize  $k$  and  $n$ .
2. Apply  $G_y$  gradient for the duration of 0.1.
3. Apply the read-out  $G_x$  gradient for the duration of 0.1.
4. Sample the value at the end of read-out duration.
5. Sample value corresponds to  $G_x, G_y$ .
6. Note that the resultant magnitude at every pixel (at time instant  $t = 0.2$ ) corresponds to the values obtained after dephasing for the duration of 0.2.

Time instant: 0.4

1. Initialize  $k$  and  $n$  (same as that of the one used in the case of time instant  $t = 0.2$ ).
2. Apply  $G_y$  gradient for the duration of 0.1.
3. Apply  $G_x$  gradient for the duration of 0.1.
4. Apply  $180^\circ$  RF pulse.
5. Apply read-out  $G_x$  gradient for the duration of 0.2.
6. Sample the value at the end of read-out duration.
7. Sample value corresponds to  $G_x, -G_y$ .
8. Note that the resultant magnitude at every pixel (at time instant  $t = 0.4$ ) corresponds to the values obtained after rephasing for the duration of 0.2 (obtained during read-out phase), after the time instant  $t = 0.2$ .
9. This is the spin-echo instant at which momentary peak has occurred.

Time instant: 0.5

1. Initialize  $k$  and  $n$  (same as that of the one used in the case of time instant  $t = 0.2$ ).
2. Apply  $G_y$  gradient for the duration of 0.1.
3. Apply  $G_x$  gradient for the duration of 0.1.
4. Apply  $180^\circ$  RF pulse.
5. Allow relaxation explicitly for the duration of 0.1. This helps to cancel out partially the phase introduced in steps 1 and 2. This step acts as the rephasing stage.
6. Apply read-out  $G_x$  gradient for the duration of 0.2. During this phase, the first 0.1 s acts as the rephasing stage to obtain the echo at 0.4. The next 0.1 s acts as the dephasing stage.
7. Sample the value at the end of read-out duration.
8. Sample value corresponds to  $G_x, -G_y$ .

9. Thus the resultant magnitude at every pixel (at time instant  $t = 0.5$ ) corresponds to the values obtained after dephasing of the resultant magnetic moment for the duration of 0.1. This is obviously lesser than the value at time instant  $t = 0.4$

The spatial matrix at three different time instances (described in the Sect. 3.2.2) are given below.

$$S_{0.2} = \begin{bmatrix} 1.00 & 5.84 & 6.11 & 2.96 & 2.79 & 2.54 & 7.93 & 6.99 & 7.59 & 1.42 & 3.42 \\ 1.00 & 5.95 & 3.56 & 2.05 & 7.00 & 5.67 & 3.33 & 1.34 & 9.14 & 1.00 & 6.02 \\ 2.29 & 4.33 & 5.73 & 4.04 & 6.44 & 4.00 & 1.65 & 5.33 & 6.53 & 1.87 & 4.68 \\ 3.74 & 3.86 & 1.89 & 7.07 & 6.67 & 4.16 & 3.63 & 2.86 & 6.67 & 5.09 & 3.75 \\ 6.69 & 3.26 & 2.86 & 3.18 & 3.72 & 6.90 & 5.76 & 6.66 & 4.41 & 1.81 & 1.45 \\ 6.06 & 2.00 & 3.53 & 4.44 & 3.94 & 2.97 & 3.45 & 4.92 & 1.00 & 7.98 & 4.58 \\ 6.16 & 4.48 & 1.97 & 3.96 & 3.87 & 6.15 & 5.39 & 7.60 & 6.42 & 3.03 & 4.04 \\ 5.40 & 4.84 & 3.27 & 3.95 & 5.90 & 6.34 & 5.56 & 5.78 & 3.33 & 2.50 & 5.27 \\ 8.72 & 6.32 & 1.00 & 5.23 & 3.13 & 4.72 & 1.00 & 1.84 & 5.00 & 3.89 & 6.11 \\ 2.18 & 2.98 & 2.55 & 3.61 & 2.00 & 2.57 & 1.86 & 2.00 & 7.09 & 1.94 & 1.00 \\ 4.24 & 2.31 & 2.45 & 5.02 & 8.00 & 6.04 & 4.45 & 2.00 & 2.09 & 2.41 & 1.55 \end{bmatrix}$$

$$S_{0.4} = \begin{bmatrix} 2.00 & 6.00 & 7.00 & 4.00 & 3.00 & 3.00 & 9.00 & 9.00 & 10.00 & 2.00 & 4.00 \\ 1.00 & 7.00 & 4.00 & 3.00 & 9.00 & 8.00 & 4.00 & 2.00 & 10.00 & 1.00 & 7.00 \\ 3.00 & 6.00 & 7.00 & 5.00 & 8.00 & 5.00 & 2.00 & 7.00 & 9.00 & 3.00 & 6.00 \\ 4.00 & 4.00 & 2.00 & 10.00 & 9.00 & 5.00 & 5.00 & 4.00 & 8.00 & 8.00 & 4.00 \\ 8.00 & 4.00 & 5.00 & 4.00 & 5.00 & 9.00 & 8.00 & 8.00 & 6.00 & 2.00 & 2.00 \\ 9.00 & 2.00 & 4.00 & 5.00 & 6.00 & 4.00 & 4.00 & 6.00 & 1.00 & 9.00 & 6.00 \\ 8.00 & 6.00 & 2.00 & 4.00 & 5.00 & 8.00 & 8.00 & 10.00 & 8.00 & 4.00 & 6.00 \\ 6.00 & 6.00 & 4.00 & 4.00 & 8.00 & 8.00 & 7.00 & 7.00 & 5.00 & 3.00 & 8.00 \\ 10.00 & 8.00 & 1.00 & 6.00 & 4.00 & 6.00 & 1.00 & 3.00 & 5.00 & 4.00 & 9.00 \\ 3.00 & 3.00 & 4.00 & 5.00 & 2.00 & 4.00 & 2.00 & 2.00 & 9.00 & 2.00 & 1.00 \\ 6.00 & 3.00 & 3.00 & 6.00 & 9.00 & 8.00 & 6.00 & 2.00 & 3.00 & 3.00 & 2.00 \end{bmatrix}$$

$$S_{0.5} = \begin{bmatrix} 1.73 & 5.96 & 6.77 & 3.72 & 2.95 & 2.88 & 8.72 & 8.44 & 9.37 & 1.85 & 3.85 \\ 1.00 & 6.71 & 3.89 & 2.74 & 8.47 & 7.37 & 3.83 & 1.88 & 9.78 & 1.00 & 6.75 \\ 2.81 & 5.56 & 6.66 & 4.75 & 7.59 & 4.74 & 1.91 & 6.55 & 8.34 & 2.69 & 5.65 \\ 3.94 & 3.96 & 1.97 & 9.23 & 8.36 & 4.78 & 4.63 & 3.68 & 7.65 & 7.22 & 3.94 \\ 7.650 & 3.80 & 4.42 & 3.78 & 4.67 & 8.45 & 7.40 & 7.65 & 5.58 & 1.95 & 1.86 \\ 8.21 & 2.00 & 3.88 & 4.85 & 5.46 & 3.73 & 3.86 & 5.72 & 1.00 & 8.74 & 5.60 \\ 7.51 & 5.60 & 2.00 & 3.99 & 4.71 & 7.52 & 7.30 & 9.35 & 7.58 & 3.75 & 5.48 \\ 5.85 & 5.70 & 3.80 & 3.99 & 7.45 & 7.55 & 6.63 & 6.67 & 4.56 & 2.87 & 7.27 \\ 9.66 & 7.55 & 1.00 & 5.80 & 3.77 & 5.66 & 1.00 & 2.68 & 5.00 & 3.97 & 8.22 \\ 2.78 & 3.00 & 3.58 & 4.64 & 2.00 & 3.61 & 1.96 & 2.00 & 8.46 & 1.98 & 1.00 \\ 5.53 & 2.82 & 2.86 & 5.73 & 8.72 & 7.47 & 5.59 & 2.00 & 2.76 & 2.85 & 1.88 \end{bmatrix}$$

Note that the elements of the matrix  $S_{0.4}$  corresponds to the local maxima occurred due to rephasing (spin echo).

### 3.2.2.1 Spinechodemonstration.m

```

siz=11;
K=0.2; %Initial phase
Mxyit=[];
gamma=2*pi*42.58*(10^6);
d1=((10^3)/(11*42.58*10^6*0.1));
d2=((10^3)/(11*42.58*10^6*0.1));
B0=1;
n=round(rand(siz,siz)*9+1);
for x=1:1:siz
    for y=1:1:siz
        for i=1:1:n(x,y)
            k(x,y,i)=rand*2-1;
        end
    end
end

KSPACE=zeros(11,11);
for i=1:1:3
    KKSPACE{i}=zeros(11,11);
end
for Gx=-5:1:5;
    for Gy=-5:1:5;
        Gx1=Gx/(10^3);
        Gy1=Gy/(10^3);
        for x=0:1:siz-1
            for y=0:1:siz-1
                for i=1:1:n(x+1,y+1)
                    phi1=k(x+1,y+1,i)*0.1*2*pi;
                    %The magnetic moments after applying positive y-gradient for the duration of
                    %0.1
                    temp1(x+1,y+1,i)=exp(-j*( (B0*gamma+gamma*y*d2*Gy1)*0.1-phi1+K));
                    %Case 1:
                    %At time instant=0.2 (refer section 3.2.2)
                    phi2=k(x+1,y+1,i)*0.1*2*pi;
                    temp2(x+1,y+1,i)=exp(-j*( (B0*gamma+d2*gamma*y*Gy1)*0.1-phi1+K))...
                    *exp(-j*( (B0*gamma+gamma*d1*x*Gx1)*0.1-phi2));

                    %Case 2:
                    %At time instant=0.4 (refer section 3.2.2)

                    phi3=k(x+1,y+1,i)*0.2*2*pi;
                    temp3(x+1,y+1,i)=exp(j*( (B0*gamma+gamma*d2*y*Gy1)*0.1-phi1+K))...
                    *exp(j*( (B0*gamma+gamma*d1*x*Gx1)*0.1-phi2))*...
                    exp(-j*( (B0*gamma+gamma*d1*x*Gx1)*0.2-phi3));

                    %Case 3:
                    %At time instant=0.8 (refer section 3.2.2)
                    phi4=k(x+1,y+1,i)*0.1*2*pi;
                    phi5=k(x+1,y+1,i)*0.2*2*pi;
                    temp4(x+1,y+1,i)=exp(j*( (B0*gamma+gamma*d2*y*Gy1)*0.1-phi1+K))...
                    *exp(j*( (B0*gamma+gamma*d1*x*Gx1)*0.1-phi2))*...
                    exp(-j*( (B0*gamma)*0.1-phi4))*exp(-j*( (B0*gamma+gamma*d1*x*Gx1)*0.2-phi5));
                end
            end
        end
    end
end

for x=0:1:siz-1
    for y=0:1:siz-1

```

```

for i=1:1:n(x+1,y+1)
    KKSPACE{1}(Gx+6,Gy+6)=KKSPACE{1}(Gx+6,Gy+6)+temp2(x+1,y+1,i);
    KKSPACE{2}(Gx+6,Gy+6)=KKSPACE{2}(Gx+6,Gy+6)+temp3(x+1,y+1,i);
    KKSPACE{3}(Gx+6,Gy+6)=KKSPACE{3}(Gx+6,Gy+6)+temp4(x+1,y+1,i);
end
end
end
end

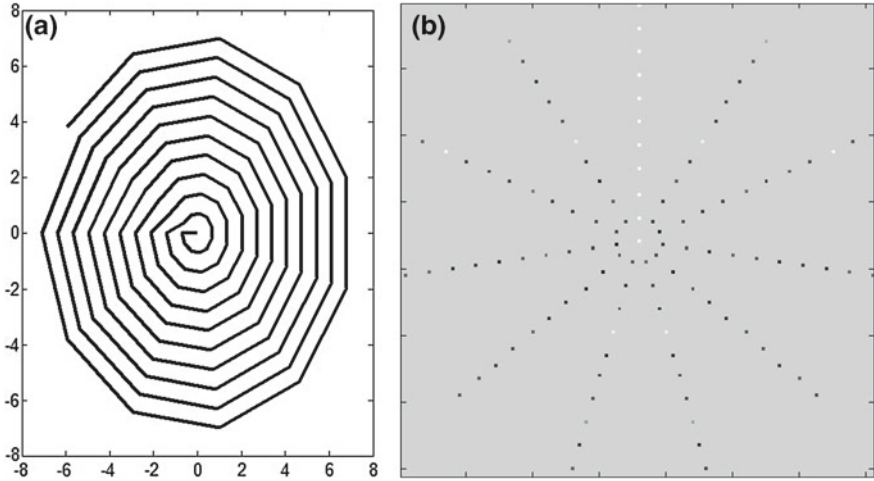
KSPACE=KKSPACE{1};
KSPACEREARRANGED=KSPACE(6:1:11,6:1:11);
KSPACEREARRANGED=[KSPACEREARRANGED KSPACE(6:1:11,1:1:5)];
KSPACEREARRANGED=[KSPACEREARRANGED; KSPACE(1:1:5,6:1:11) KSPACE(1:1:5,1:1:5)];
SPINECHO{1}=abs(iff2(KSPACEREARRANGED));
%Note that what we have reconstructed f(x,y)

for i=2:1:3
    KSPACE=KKSPACE{i};
    KSPACEREARRANGED=KSPACE(6:1:11,6:1:11);
    KSPACEREARRANGED=[KSPACEREARRANGED KSPACE(6:1:11,1:1:5)];
    KSPACEREARRANGED=[KSPACEREARRANGED; KSPACE(1:1:5,6:1:11) KSPACE(1:1:5,1:1:5)];
    RECONSTRUCTED=abs(iff2(KSPACEREARRANGED));
    %But note that what we reconstructed is f(x,-y) and not f(x,y).
    %f(x,y) is reconstructed as follows.
    FINALRECONSIMAGE=zeros(11,11);
    FINALRECONSIMAGE(:,1)=RECONSTRUCTED(:,1);
    for p=1:1:11
        for q=2:1:11
            FINALRECONSIMAGE(p,q)=RECONSTRUCTED(p,13-q);
        end
    end
    SPINECHO{i}=FINALRECONSIMAGE;
end
end

```

### 3.3 Illustration on the Steps Involved in Obtaining the $T_2$ MRI Image Using Polar Scanning

1. Initialize  $n$  and  $k$ .
2. Apply  $G_x$  and  $G_y$  gradient simultaneously for the duration of 0.1 corresponding to the controlling parameter  $r$  and  $\theta$  i.e.  $G_x = r\cos(\theta)$  and  $G_y = r\sin(\theta)$ .
3. Apply  $180^\circ$  pulse for the duration of 0.3.
4. Allow the resultant magnetic moment to dephase for the duration of 0.3.
5. Apply the  $G_x$  and  $G_y$  gradient simultaneously for the duration of 0.2 as the read-out phase. Sample the value at the end of the read-out phase. This corresponds to  $G_x = r\cos(\theta)$  and  $G_y = r\sin(\theta)$ .
6. Note that the steps used in this technique are exactly same as that of Cartesian spin-echo based  $T_2$  MRI. The difference is, in this case K-space is scanned in the polar fashion as shown in the Fig. 3.6.
7. This is repeated for specific range of  $r$  (depends upon on the resolution required for the image slice) and  $\theta$  ranges from 0 to  $\pi$ .
8. K-space data  $F(G_x, G_y) = F(r\cos(\theta), r\sin(\theta))$  is rewritten as  $G(r, \theta)$ .



**Fig. 3.6** **a** Illustrating the order in which the K-space is scanned; **b** sample polar scanned K-space

9. Apply inverse fourier transformation in polar form to obtain  $f(x, y)$  (refer Sect. 3.3.1).
10. For checking the validity of reconstruction from polar data,  $k$  is chosen as zero matrix (No dephasing) for illustration. This helps to obtain proton-density image (n-matrix) as the reconstructed image.

### 3.3.1 Reconstructing $f(x, y)$ from $G(r, \theta)$

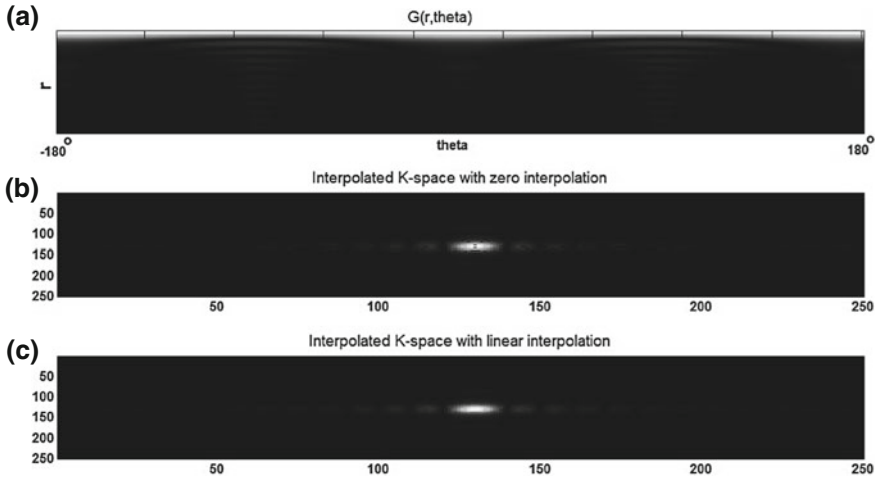
There are two major techniques to obtain  $f(x, y)$  from  $G(r, \theta)$  namely back-projection technique and the interpolation technique as described below. The interpolation technique is used for illustration purpose.

#### 3.3.1.1 Back-Projection Technique

The generalized formula to reconstruct  $f(x, y)$  from  $G(r, \theta)$  is as shown below.

$$f(x, y) = \int_{-\pi}^{\pi} \int_0^{\infty} G(r, \theta) \exp^{j2\pi(x\cos\theta + y\sin\theta)r} |J| dr d\theta \quad (3.2)$$

1. It is noted  $G(r, \theta)$  is the fourier transformation of the radon transformation  $R(l, \theta)$  for the specific constant  $\theta$ .
2. It is possible to get back  $f(x, y)$  by getting  $R(l, \theta)$  using inverse fourier transformation (refer Sect. 1.2), followed by back-projection technique.



**Fig. 3.7** a Fourier transformation of the radon transformation (column-wise) for the specific  $\theta$ ; b interpolated K-space with zero interpolation c Interpolated K-space with linear interpolation

### 3.3.1.2 Interpolation Technique

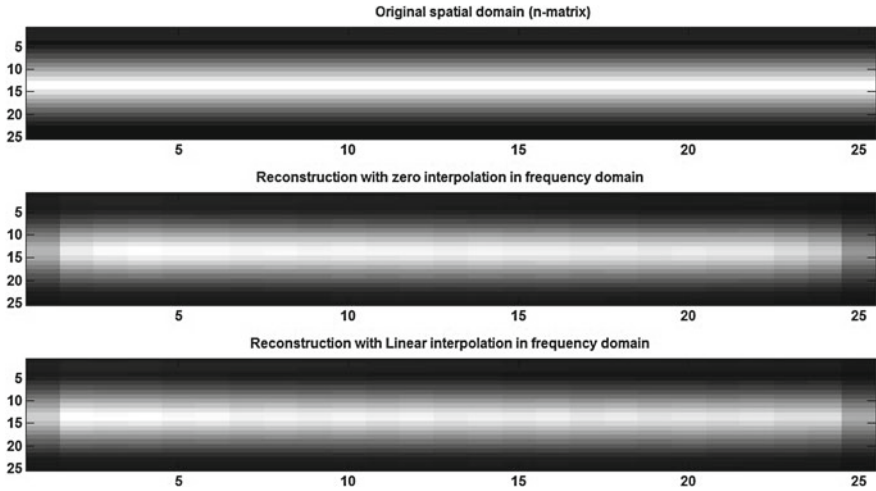
1. Interpolate to obtain the  $F(G_x, G_y)$  (with more number of sample points in frequency domain) from polar scanned  $F(G_x \cos(\theta), G_y \sin(\theta))$ .
2. Apply inverse IFFT2 of  $F(G_x, G_y)$  to obtain  $f(x, y)$ .
3. The quality of the MRI image can be improved by increasing the resolution of the variable  $\theta$ .
4. Interpolation in the frequency domain increases the number of samples obtained in 2D frequency domain between  $-\pi$  to  $\pi$  in both the directions.
5. Interpolation in the frequency domain will not increase the spatial resolution in the spatial domain. But it is equivalent to padding zeros in the spatial domain in both the directions.
6. Hence after obtaining IFFT2 of the interpolated K-space, collect the  $25 \times 25$  matrix from the top-left corner to obtain the image (refer Fig. 3.7).
7. The original and the reconstructed image (n-matrix) using zero interpolation and the liner interpolation are illustrated in the Fig. 3.8.

### 3.3.1.3 Spinechopolar.m

```

siz=25;
K=0.2; %Initial phase
Mxyit=[];
%instead of 59 ,2559
gamma=2*pi*42.58*(10^6);
d1=((10^3)/(25*42.58*10^6*0.1));
d2=((10^3)/(25*42.58*10^6*0.1));
%d1=1;
    
```





**Fig. 3.8** Illustration of the original and the reconstructed spatial domain (proton-density) from the polar scanned K-space  $G(r, \theta)$

```

%d2=1;
B0=1;
n=ones(3,25)*2;
for i=1:1:10
    n=[n;ones(1,25)*i];
end
for i=10:-1:1
    n=[n;ones(1,25)*i];
end
n=[n;ones(2,25)*1];
POS=[];
for x=1:1:siz
    for y=1:1:siz
        for i=1:1:n(x,y)
            % k(x,y,i)=rand*2-1; %With dephasing
            k(x,y,i)=0; %Just to check the validity of the program
        end
    end
end
end
KSPACE=zeros(101,180);
for r=0:(13/100):13
    r1=round(r)*(100/13)+1;
    for theta=-pi:(2*pi)/179:pi
        theta1=round((theta+pi)*(179/(2*pi))+1);
        Gx=r*cos(theta) ;
        Gy=r*sin(theta);
        Gx1=Gx/(10^3);
        Gy1=Gy/(10^3);
        POS=[POS;
            Gx Gy];

        temp=0;
        for x=0:1:24
            for y=0:1:24
                for i=1:1:n(x+1,y+1)
                    phi1=k(x+1,y+1,i)*0.1*2*pi;
                    %x-Gradient and y-Gradient are applied simultaneously for the duration of
                    %0.1

```

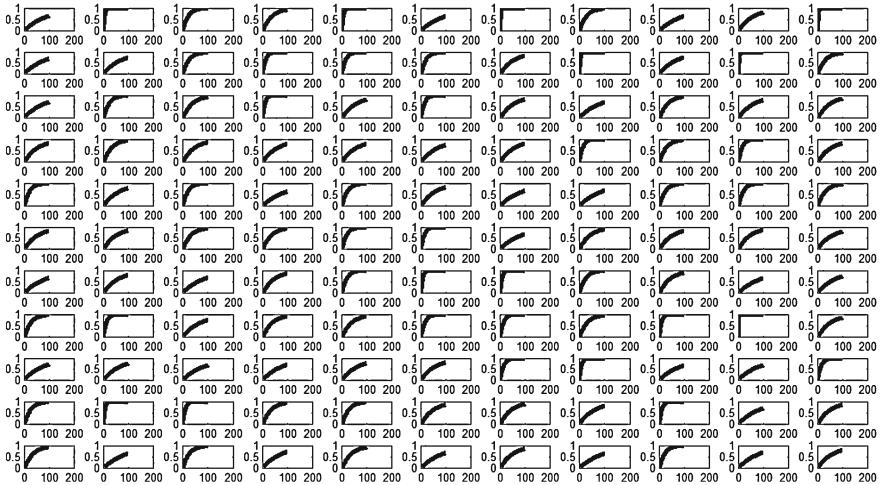
```

temp1(x+1,y+1,i)=exp(-j*((B0*gamma+gamma*y*d2*Gy1)*0.1-phi1+K))...
*exp(-j*((B0*gamma+gamma*x*d1*Gx1)*0.1-phi1+K));
%Apply 180 degree phase shift and allow it to dephase for 0.3 duration
phi2=k(x+1,y+1,i)*0.3*2*pi;
temp2(x+1,y+1,i)=exp(j*((B0*gamma+gamma*y*d2*Gy1)*0.1-phi1+K))*...
exp(j*((B0*gamma+gamma*x*d1*Gx1)*0.1-phi1+K))*...
exp(-j*((B0*gamma)*0.3-phi2));
%Positive x-Gradient and y-Gradient are applied simultaneously for the duration of
%0.2 as the read-out time dphase. At the end of the read-out phase,the
%k-space is sampled corresponding to (r,theta)
phi3=k(x+1,y+1,i)*0.2*2*pi;
temp3(x+1,y+1,i)=exp(j*((B0*gamma+gamma*y*d2*Gy1)*0.1-phi1+K))*...
exp(j*((B0*gamma+gamma*x*d1*Gx1)*0.1-phi1+K))*...
exp(-j*((B0*gamma)*0.3-phi2))*exp(-j*((B0*gamma+gamma*y*d2*Gy1)*...
0.2-phi3))*exp(-j*((B0*gamma+gamma*x*d1*Gx1)*0.2-phi3));
end
end
end
for x=0:1:siz-1
    for y=0:1:siz-1
        for i=1:1:n(x+1,y+1)
            KSPACE(r1,theta1)=KSPACE(r1,theta1)+temp3(x+1,y+1,i);
        end
    end
end
end
%Reconstruction using second technique(refer section 2.11.2)
F=KSPACE;
POS=round((POS+14)*10);
%Number of samples in frequency domain is increased as 250x250 from 25x25
%and zero interpolated.
A=zeros(250,250);
r1=1;
theta1=1;
for i=1:1:length(POS)
    F(r1,theta1)
    A(POS(i,1),POS(i,2))=F(r1,theta1) ;
    theta1=theta1+1;
    if(theta1==181)
        r1=r1+1;
        theta1=1;
    end
end
end
figure
A=fftshift(A);
imagesc(abs(A))
C1=abs(iff2(A));

%Increase the number of samples in frequency domain from 25x25 to 250x250
%and apply Linear interpolation,considering 8 neighbours of every pixel.
B=A;
[p,q]=find(B==0);
C=[zeros(1,250);B;zeros(1,250)];
C=[zeros(252,1) C zeros(252,1)];
p=p+1;q=q+1;
D=C;

for i=1:1:length(p)
    vector=[C(p(i)-1,q(i)-1) C(p(i)-1,q(i)+1) C(p(i)+1,q(i)-1) C(p(i)+1,q(i)+1) ...
            C(p(i),q(i)-1) C(p(i),q(i)+1) C(p(i)+1,q(i)) C(p(i)+1,q(i))];
    D(p(i),q(i))=mean(vector);
end
end
B=D(2:1:251,2:1:251);

```



**Fig. 3.9** Illustration of the increase in the magnitude of the resultant longitudinal magnetic moment at every pixel (of the  $11 \times 11$  image slice) due to natural longitudinal relaxation (spin–lattice interaction based)

```

figure
imagesc(abs(B))
C2=abs(iff2(B));

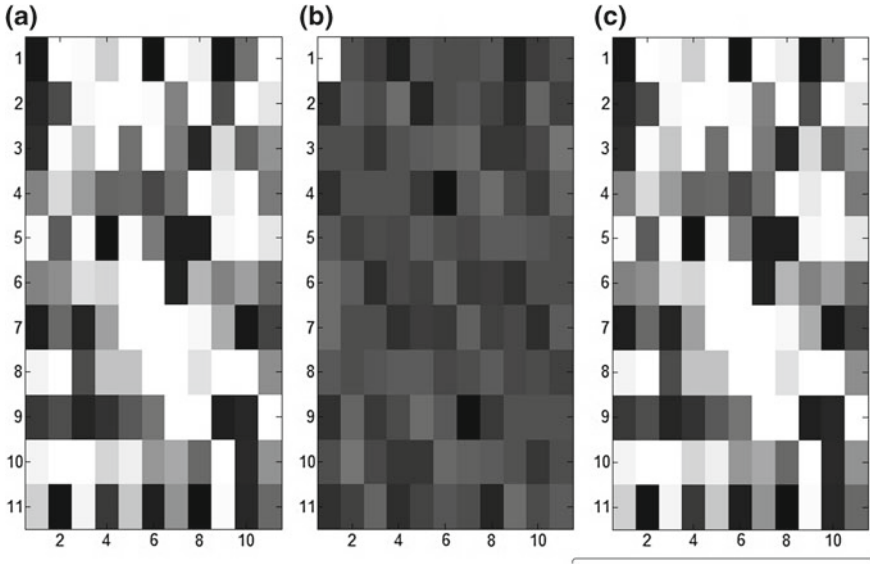
figure
colormap(gray)
subplot(3,1,1)
imagesc(n)
title('Original')
subplot(3,1,2)
imagesc(C1(1:1:25,1:1:25))
title('With zero interpolation in frequency domain')
subplot(3,1,3)
title('With Linear interpolation in frequency domain')
imagesc(C2(1:1:25,1:1:25))

```

### 3.4 Illustration on the Steps Involved in Obtaining the $T_1$ MRI Image

The following steps are followed to obtain the  $T_1$  image.

1. Select the image slice using z-gradient. Apply refocussing gradient to compensate the phase introduced due to z-gradient.
2. Apply  $90^\circ$  pulse to obtain the transverse component.
3. The transverse component at different pixel starts dephasing due to spin–spin interaction (described by the factor  $T_2(x, y)$ ).



**Fig. 3.10** **a** Actual magnitude of the transverse component immediate after applying the second  $90^\circ$  pulse; **b** K-space; **c** reconstructed image, corresponding to the time constant  $T_1$

4. After approximately 3 times  $T_2(x, y)$ , transverse component is almost zero. During this duration, the longitudinal component gradually increases (refer Fig. 3.9). This is described by the factor  $T_1(x, y)$ .
5. At this moment, apply another  $90^\circ$  degree pulse. This is required to obtain the transverse component. Now the magnitude of the magnetic moment in the individual pixel of the image slice are different at various locations (refer Fig. 3.10a).
6. Apply  $G_y$  gradient for the duration of 0.0000000009.
7. Apply  $-G_x$  gradient for the duration of 0.0000000009.
8. Apply  $G_x$  gradient for the duration of 0.0000000018. During this phase, after the time duration of 0.0000000009, there is the cancellation of phase introduced due to  $G_x$ . This is known as Gradient echo. This helps to synchronize the hardware to sample at the particular instant during real time to choose the proper position in the K-space.
9. Sample the real and imaginary component of the signal  $s(t)$  to obtain the sample of the K-space at  $(G_x, G_y)$ .
10. Proper scaling factor is used so that the final basis (to perform IFFT2) look like the standard form  $e^{-\frac{j2\pi x G_x}{11}} e^{-\frac{j2\pi y G_y}{11}}$ . This is the process of discretization.
11. The above steps (2–10) are repeated for the complete scan in the K-space. For every time, we have to wait for the longitudinal component to reach maximum before applying  $90^\circ$  RF pulse. (This is illustrated by repeating the steps 1–5 by varying the values of  $G_x$  and  $G_y$  ranging from  $-5$  to  $5$  with the interval of 1).

12. In the earlier illustrations ( $T_2$  and proton-density imaging), we have considered the proton-density at every pixel(number of magnetic moments). But in this case, the illustration is done in the macro level at every pixel.
13. The actual magnitude of the transverse component immediate after applying the second  $90^\circ$  pulse, K-space and the corresponding the reconstructed image are shown in the Fig. 3.10.

### 3.4.1 *t1.m*

```

siz=11;
K=0.2; %Initial phase
gamma=2*pi*42.58*(10^6);
d1=((10^3)/(11*42.58*(10^6)*0.000000009));
d2=((10^3)/(11*42.58*(10^6)*0.000000009));
t1=1;
B0=1
%Mz(x,y) is the longitudinal component at (x,y).
T1=rand(11,11);
for tb=0:0.01:1;
    t1=t1+1;
    for x=1:1:11
        for y=1:1:11
            Mz{x,y}(t1)=(1-exp(-tb/T1(x,y)))*B0;
        end
    end
end
end
i=1;
%snapshot
for x=1:1:11
    for y=1:1:11
        subplot(11,11,i)
        plot(Mz{x,y})
        i=i+1;
    end
end
%At tb=1 (at which the original transverse component due
%to dephasing is almost zero),90 degree pulse is again applied.
%Now the transverse component at different
%pixel in the image slice have different magnitude.
for x=1:1:11
    for y=1:1:11
        M(x,y)=Mz{x,y}(101);
    end
end
end
%M is the magnitude of the individual transverse component at every pixel
%immediate after applying the 90 degree pulse (at t=0).

%Sampling the k-space at Gx,Gy.
temp1=0;
temp2=0;
temp3=0;
KSPACE=zeros(11,11);
for Gx=-5:1:5;
    for Gy=-5:1:5;
        Gy1=Gy/(10^3);%This is to indicate that B0 is high compared with y-gradient
    end
end

```

```

Gx1=Gx/(10^3);%This is to indicate that B0 is high compared with x-gradient
for x=0:1:siz-1
for y=0:1:siz-1
%The magnetic moments after applying y-gradient for the duration of
%0.000000009
temp1=temp1+M(x+1,y+1)*exp(-j*((B0*gamma+gamma*y*d2*Gy1)*0.000000009+K));
%The magnetic moments after applying negative x-gradient for the duration of
%0.000000009 after applying the y-gradient for the duration of
%0.000000009
temp2=temp2+M(x+1,y+1)*exp(-j*((B0*gamma+d2*gamma*y*Gy1)*0.000000009+K))...
*exp(-j*((B0*gamma-gamma*d1*x*Gx1)*0.000000009));
%The magnetic moments after applying x-gradient for the duration of
%0.0000000018 after applying the y-gradient for the duration of
%0.000000009 and negative x-gradient for the duration of 0.000000009
temp3=temp3+M(x+1,y+1)*exp(-j*((B0*gamma+gamma*d2*y*Gy1)*0.000000009+K))...
*exp(-j*((B0*gamma-gamma*d1*x*Gx1)*0.000000009))*...
exp(-j*((B0*gamma+gamma*d1*x*Gx1)*0.000000018));
end
end
KSPACE(Gx+6,Gy+6)=temp3;
temp1=0;
temp2=0;
temp3=0;
end
end
KSPACEREARRANGED=KSPACE(6:1:11,6:1:11);
KSPACEREARRANGED=[KSPACEREARRANGED KSPACE(6:1:11,1:1:5)];
KSPACEREARRANGED=[KSPACEREARRANGED; KSPACE(1:1:5,6:1:11) KSPACE(1:1:5,1:1:5) ];
RECONSTRUCTED=ifft2(KSPACEREARRANGED)
figure
subplot(1,3,1)
colormap(gray)
imagesc(M)
subplot(1,3,2)
colormap(gray)
imagesc(log(abs(KSPACEREARRANGED)))
subplot(1,3,3)
colormap(gray)
imagesc(abs(RECONSTRUCTED))

```

# Chapter 4

## Medical Image Processing

### 4.1 Summary on the Various Medical Imaging Techniques

1. X-ray image: X-ray is transmitted through the body and are captured on the other side of the body. The attenuation introduced in the ray helps in obtaining the image. There are different modalities used in X-ray imaging. Example: Angiography, Neuroradiology.
2. Computed Tomography (CT): This is identical to X-ray. But in this case, the sequence of images are captured by rotating the image slice for  $180^\circ$  with the resolution of  $\theta_{\text{res}}$  degree. This helps to capture the particular slice of the human body. Refer Chap. 1 for further details.
3. Magnetic Resonance Imaging (MRI): In this case, human body is excited with the strong external magnetic field to obtain the resultant magnetic moment at every voxel to align in the identical direction. These magnetic moments are disturbed with the external RF excitation and the transverse component of the magnetic moments are captured. Based on the methodology, MRI is capable of capturing the characteristics of the proton-density (PD image), spin–spin interaction between the magnetic moments (T2 image), spin–lattice interaction of the magnetic moments (T1 image). Refer Chap. 2 for further details.
4. Ultrasound image: The ultrasound waves are transmitted through the human body and the reflected wave (due to reflection and the scattering) is recorded to obtain the image. There are different modalities used in Ultrasound imaging systems. Example: Doppler imaging.
5. Nuclear medicine image: The appropriate radioactive substances are introduced to the body and the emission of the gamma rays are recorded to obtain the image. The DSP's used in CT images are used in this imaging technique also. The difference is, in this case gamma rays emitted due to the introduced radioactive substances are used instead of x-rays. Example: Planar scintigraphy, Positron emission tomography (PET), Single photon emission computed tomography (SPECT).

## 4.2 Image Enhancement

The images obtained using the medical imaging technique are not usually suitable for direct display. There is the need to process the image so that the required portion of the image are highlighted properly for better visibility.

### 4.2.1 Logarithmic Display

When the range of gray values available in the image is very large (Example: The spectrum of the image), linear colormap will not convey the complete information (refer Fig. 4.4a).

If the logarithmic of the image values are taken and is displayed using the linear colormap, we get the enhanced look of the image, as shown in the Fig. 4.4b.

### 4.2.2 Non-Linear Filtering

The non-linear filter like median filter is used to filter the spot-noise in the image as described below.

1. Create the zero matrix  $Z$  with the size of the input image  $I$  to be filtered.
2. For every  $3 \times 3$  (or more) overlapping subblocks of the image  $I$ , compute the median of all the pixel values to obtain the value  $P$ .
3. The position of the center pixel of the subblock is noted as  $(m, n)$ .
4. The pixel  $(m, n)$  of matrix  $Z$  is replaced with  $P$ . The image thus obtained is median filtered image. The image with spot noise and the corresponding filtered image using median filter is illustrated in the Fig. 4.1.

### 4.2.3 Image Substraction

During the treatment of the diseases (that involves medical imaging), the progress in the treatment can be identified by subtracting the image captured before and after treatment. This highlights the progress in the treatment. There are certain imaging technique where indicator is used. The difference between the image captured before and after the injection of indicator helps in enhanced visibility. The difference between the successive frame of the MRI video (26 frames) helps in highlighting the function of the organ (refer Fig. 4.2).



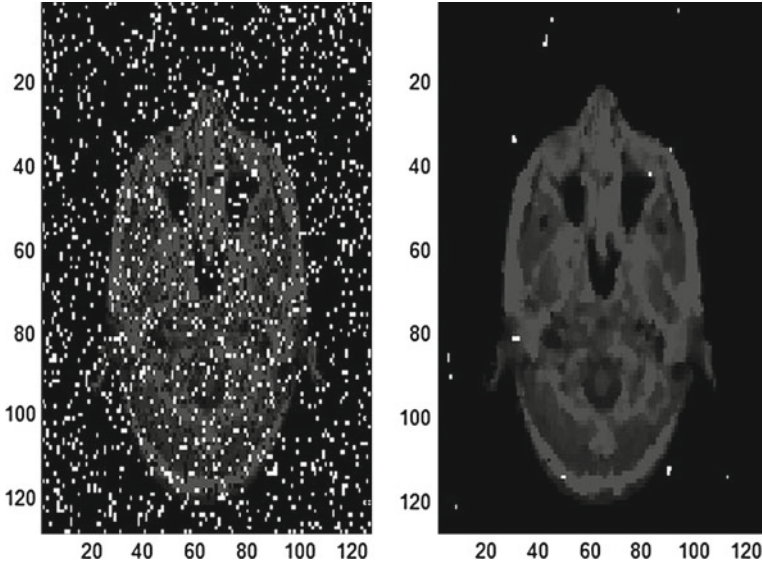


Fig. 4.1 MRI image with impulsive noise and the corresponding filtered image using median filter

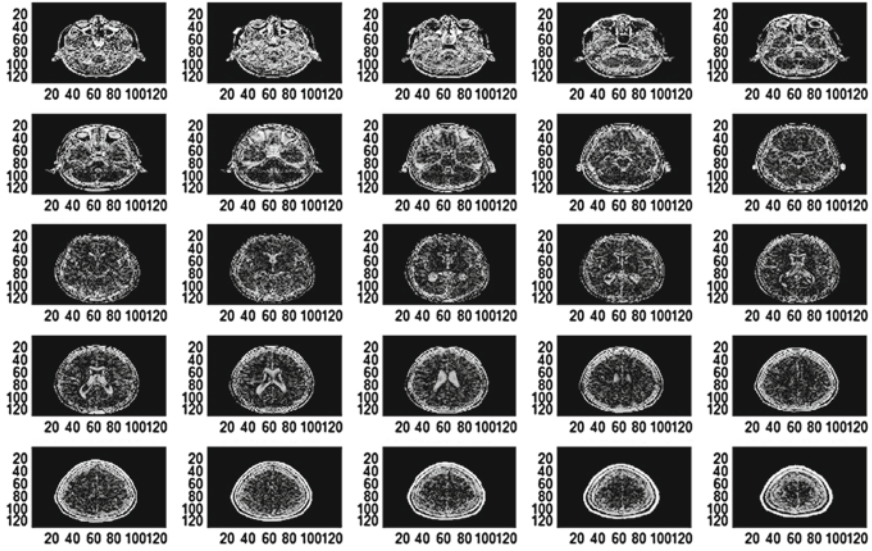
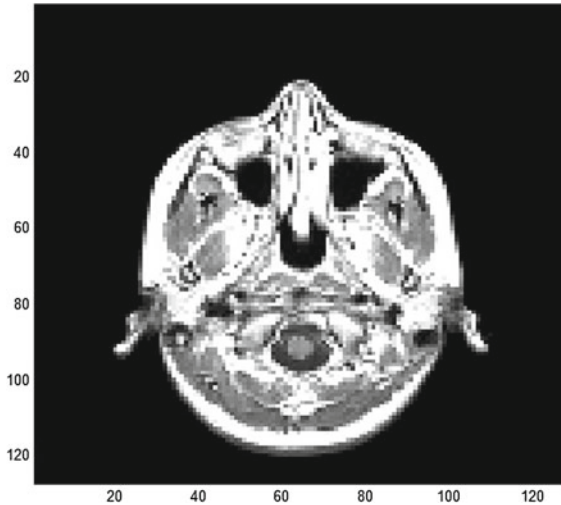


Fig. 4.2 Difference image in the logirthmic scale

**Fig. 4.3** Original MRI image subjected to linear filtering



#### ***4.2.4 Linear Filterering and the Hankel Transformation***

Gradual change in the pixel value gives the low frequency content of the image.

The sudden change in the pixel values give the high frequency content of the image. Based on the requirement, we can view the image by using linear filtering as demonstrated below.

1. Consider the MRI image as shown in the Fig. 4.3.
2. The magnitude spectrum of the MRI image and its logirthmic view (with origin in the middle) is shown in the Fig. 4.4.
3. The logirthmic spectrum of the image after passing through the filter and its corresponding spatial domain images are shown in the Fig. 4.5.
4. It is observed that the Fig. 4.5b consists of low frequency content of the image (overall information, gradual change) and Fig. 4.5d consists of high frequency content of the image (detail information).

##### **4.2.4.1 Hankel Transformation**

The typical magnitude response (spectrum) of the ideal low pass filter is as shown in the Fig. 4.6. It is observed that the filter is circularly symmetric (we get the identical spectrum even after rotating the spectrum image with an angle  $\theta$ ). This property is exploited to obtain the hankel transformation and are used to design any circularly symmetric filter as described below.

1. As the spectrum is circularly symmetric, we consider the values along the y-axis from 0 to  $\infty$  and are represented as  $F(q)$ . The hankel transformation



**Fig. 4.4** Original spectrum and the logarithmic spectrum of the MRI image (with origin) in the middle

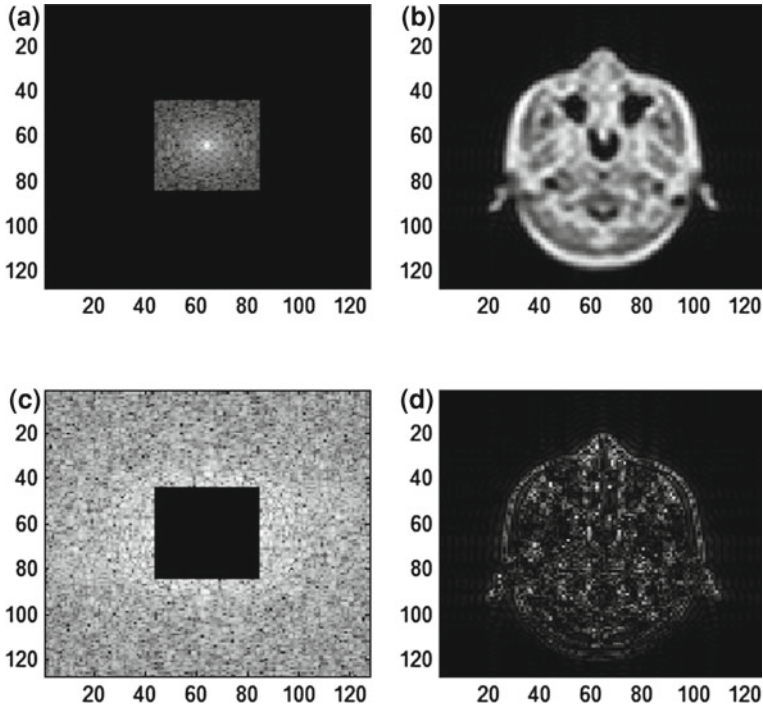
(refer Appendix) of the function  $F(q)$  is given as

$$f(r) = 2\pi \int_0^{\infty} q F(q) J_0(2\pi qr) \tag{4.1}$$

where,

$$J_0(q) = \frac{1}{\pi} \int_0^{\pi} \cos(q \sin(\phi)) \tag{4.2}$$

2. Thus the hankel transformation is applied to the function  $F(q)$  to obtain  $f(r)$ .
3. The values of  $F(r)$  are treated as the polar representation of the spatial domain  $F(r, \theta)$  for all the angle  $\theta$ .
4. For the 2D-filter specification shown in the Fig.4.6, the spatial domain of filter obtained using the direct method (2D-IDFT) and the hankel transformation are



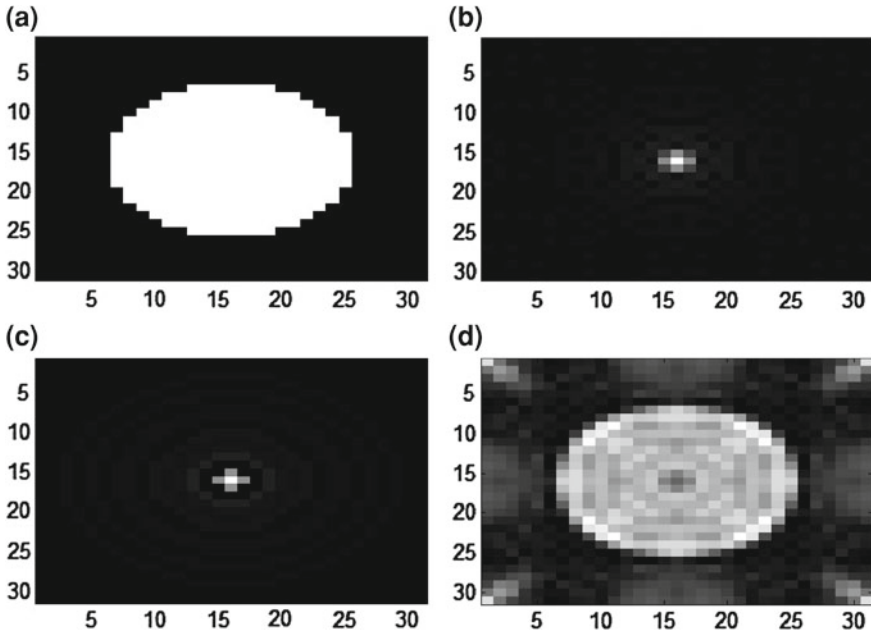
**Fig. 4.5** **a** Logarithmic low pass spectrum. **b** Image corresponding to low pass spectrum. **c** Logarithmic high pass spectrum. **d** Image corresponding to high pass spectrum

shown in the figure. It is noted that the spatial domain obtaining using the hankel transformation requires lesser number of computation to obtain the spatial domain.

5. Note that in case of continuous hankel transformation, the 2D-filter specification is perfectly circular symmetric and hence 2D-FFT consists of only the real components.
6. Ideally (1) and (4) subplots of the Fig. 4.6 must be identical. The variation is due to the imperfection in the circular symmetry while realizing in discrete form.

#### 4.2.4.2 hankeltransformation.m

```
%Demonstration of Hankel transformation
A=zeros(31,31);
a=ones(1,15);
for r=0:1:9
for theta=0:(2*pi)/360:(2*pi);
x=round(r*cos(theta));
y=round(r*sin(theta));
A(x+16,y+16)=a(r+1);
```



**Fig. 4.6** **a** Circular symmetric filter with origin as the center. **b** Corresponding spatial domain (magnitude) obtained using 2D-IDFT. **c** Spatial domain (magnitude) obtained using Hankel transformation. **d** Magnitude of 2D-FFT of spatial domain obtained using Hankel transformation

```

end
end
B=ifft2(A);
B=fftshift(B);
%The above 2D-IDFT is realized using Hankel transformation
%as described
%below.
F=[];
for i=16:1:31
    F=[F A(i,16)];
end
s=0;
for q=0:1:15
    for r=1:1:14
        s=s+2*pi*2*r*F(r+1)*Jbessel(2*pi*q*r/31);
    end
    s=s+F(16)*Jbessel(2*pi*q*15/31)
    f(q+1)=s;
    s=0;
end

```

```

C=zeros(31,31);
for r=0:1:15
    for theta=0:(2*pi)/360:2*pi;
        x1=round(r*cos(theta));
        y1=round(r*sin(theta));
        C(x1+16,y1+16)=f(r+1);
    end
end
c2=fft2(C);
c31=fftshift(c2);

figure
colormap(gray)
subplot(2,2,1)
imagesc(abs(A))
title(' (1) ')
subplot(2,2,2)
imagesc(abs(B))
title(' (2) ')
subplot(2,2,3)
imagesc(abs(C))
title(' (3) ')
subplot(2,2,4)
imagesc(abs(c31))
title(' (4) ')

```

### 4.2.5 Histogram Equalization

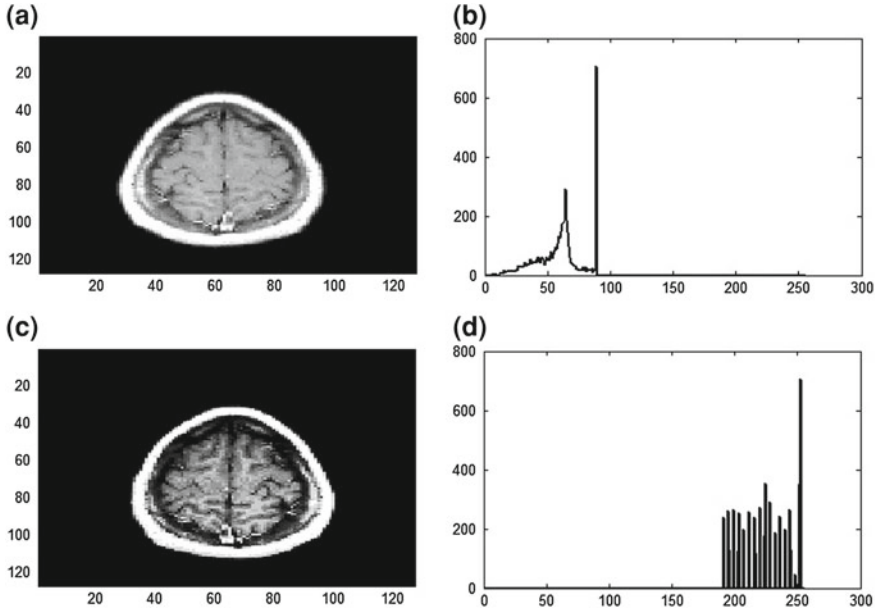
The histogram of the gray image is viewed as the probability mass function of the discrete random variable  $X$ . Let the number of levels of the random variable  $X$  be 0–255. Let the probability mass function of the original image be represented as  $P(X_i)$ . The corresponding cumulative distribution is represented as  $F_X(n) = \sum_{i=0}^n P(X_i)$ . Let the random variable of the image after histogram equalization be represented as  $Y$ . Thus the histogram equalization is the process of obtaining the transformation  $Y = g(X)$  such that the cumulative distribution of random variable  $Y$  is  $F_Y(n) = n$ . The transformation is obtained as follows.

$$F_Y(n) = P(Y \leq n) = P(g(X) \leq n) = P(X \leq g^{-1}(n)) = F_X(g^{-1}(n)) = n. \quad (4.3)$$

$$\Rightarrow F_X(g^{-1}) = 1 \quad (4.4)$$

$$\Rightarrow F_X(\alpha) = g(\alpha) \quad (4.5)$$

Thus the transfer function  $g(\alpha)$  is the cumulative distribution function of the original image itself  $F_X(\alpha)$ . The original image and the corresponding histogram equalized



**Fig. 4.7** **a** Original image. **b** Histogram corresponding to **a**. **c** Histogram equalized image. **d** Histogram corresponding to **c**

image is illustrated in the Fig.4.7. Note that the value of the mass function for the background color is made zero in the histogram plot.

### 4.2.6 Histogram Specification

It is possible to obtain the transformation  $Y = g(X)$  for the arbitrary cumulative distribution functions  $F_Y(y)$  (instead of uniform distribution as mentioned in histogram equalization) as described below. This is known as histogram specification.

1. Identify the transfer function  $Z = g_1(X)$  that maps the random variable  $X$  to intermediate random variable  $Z$ , such that  $F_Z(z) = z$ .
2. Identify the transfer function  $Z = g_2(Y)$  that maps the random variable  $Y$  to intermediate random variable  $Z$ , such that  $F_Z(z) = z$ .
3. Given the arbitrary value of the matrix  $\alpha$ , identify the corresponding value for  $Z$  using  $Z = g_1(X)$ . (say  $\beta$ ).
4. Using the function  $Y = g_2^{-1}(Z)$ , identify the value of  $Y$  corresponding to  $Z = \beta$ .
5. Repeat this for all values in the original image matrix described by the random variable  $X$  to obtain the image matrix described by the random variable  $Y$ , using the intermediate random variable  $Z$ . Thus the histogram equalized image is obtained.

### 4.2.6.1 histogramequalization.m

```

%Histogram equalization
load mri
A=D(:,:,:,25);
A=A(1:1:127,1:1:127);
B=histeq(A)';
figure
colormap(gray)
subplot(2,2,1)
imagesc(A)
subplot(2,2,2)
plot(imhist(A))
a=imhist(A);
a(1)=0;
plot(a)
subplot(2,2,3)
B=imrotate(B,-90);
imagesc(B)
subplot(2,2,4)
b=imhist(B)
b(187)=0;
plot(b)

```

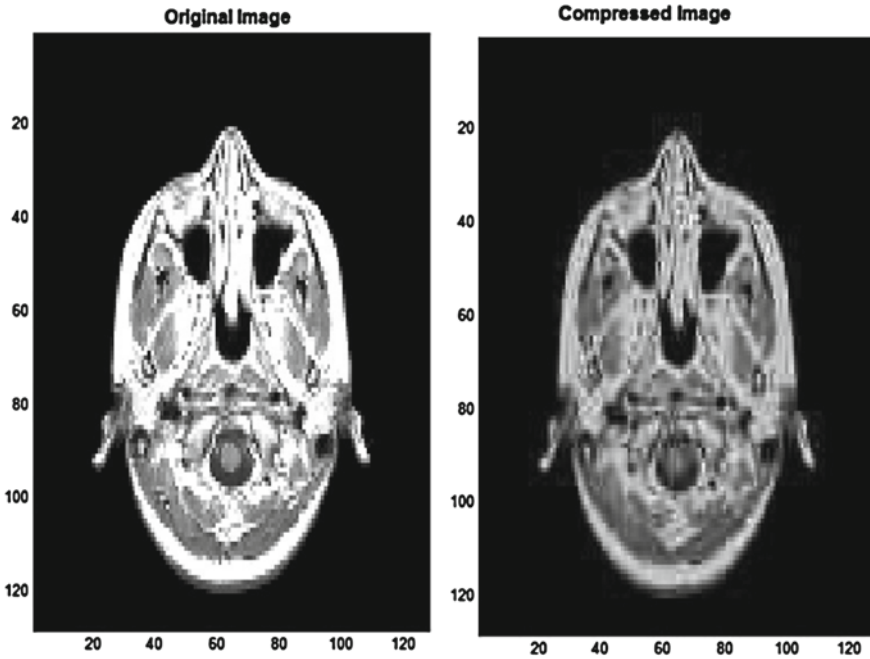
## 4.3 Image Compression

### 4.3.1 Discrete Cosine Transformation (DCT)

The following steps are followed to compress the image using DCT.

1. Subtract 128 from all the pixel values to obtain the DC component 0.
2. Divide the image into subblock of size  $8 \times 8$ .
3. Each subblock is subjected to 2D-DCT.
4. Divide each sub-block with the quantization mask (JPEG Mask) and round it. We get more number of zeros. The quantized data from each subblock is collected in the zig-zag fashion and are stored using run-length and Huffmann coding.
5. Hence image is compressed using 2D-DCT. Refer Fig. 4.8 to view the original and the corresponding compressed image obtained using 2D-DCT ( Figs. 4.9 and 4.10).





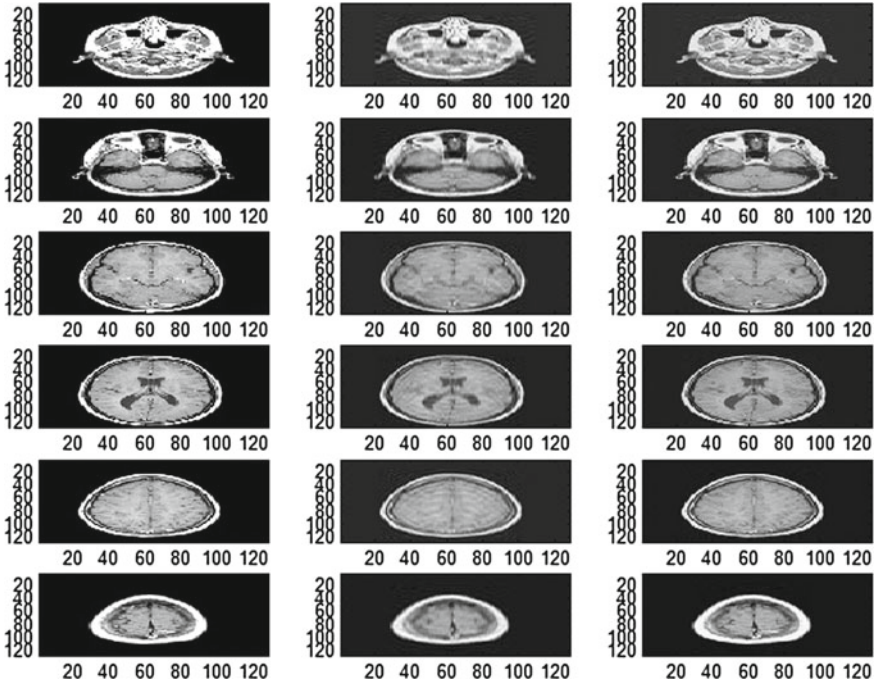
**Fig. 4.8** Illustration of the original image and the corresponding compressed image obtained using 2D-DCT

#### 4.3.1.1 imagecompusingdct.m

```
load mri
D=double(D);
V=D(:,:,1,1);
V=double(V);
V1=blkproc(V,[8 8],'dctt(x)');
V2=blkproc(V1,[8 8],'idctt(x)');
figure
colormap(gray)
subplot(1,2,1)
imagesc(abs(V))
subplot(1,2,2)
imagesc(abs(V2))
```

#### 4.3.1.2 dctt.m

```
function [res]=dctt(x)
%Function called by the imagecompusingdct.m file
x=x-128;
x=dct2(x);
JPEGQ=[16 11 10 16 24 40 51 61;12 12 14 19 26 58 60 55;14 13
```



**Fig. 4.9** a Original image. b 1:5.12 Compressed image using KL-Transformation. c 1:2.56 Compressed image using KL-Transformation

```

16 24 40 57 69 56;...
14 17 22 29 51 87 80 62;18 22 37 56 68 109 103 77;24 35
55 64 81 104 113 92;...
49 64 78 87 103 121 120 101;72 92 95 98 112 100 103 99];
res=round(x./JPEGQ);

```

**4.3.1.3 idctt.m**

```

function [res]=idctt(x)
%Function called by the imagecompusingdct.m file
JPEGQ=[16 11 10 16 24 40 51 61;12 12 14 19 26 58 60 55;14 13
16 24 40 57 69 56;...
14 17 22 29 51 87 80 62;18 22 37 56 68 109 103 77;24 35 55
64 81 104 113 92;...
49 64 78 87 103 121 120 101;72 92 95 98 112 100 103 99];
x=x.*JPEGQ;
x=idct2(x);
res=x+128;

```

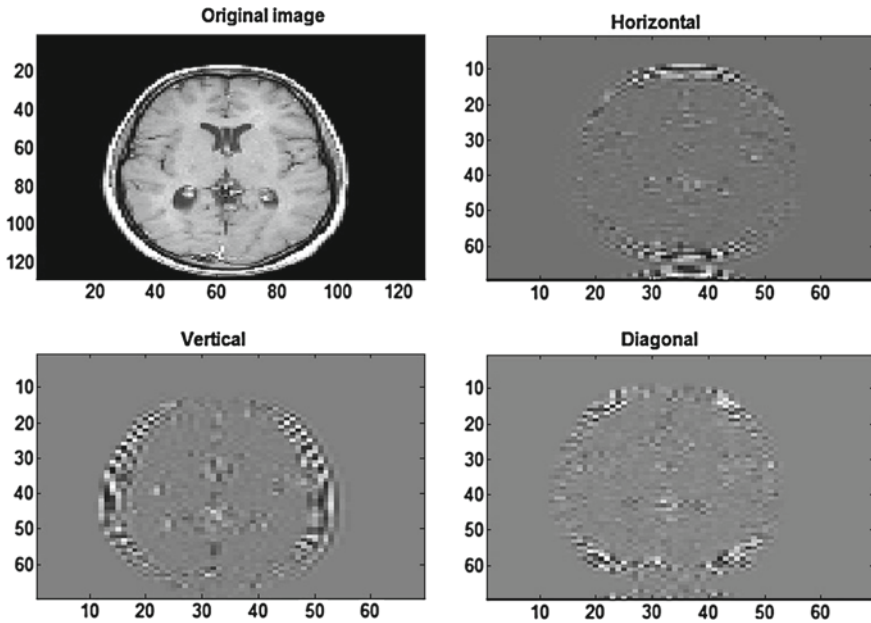


Fig. 4.10 Original image and the first level wavelet decomposed images using *db6* filter

### 4.3.2 Using KL-Transformation

1. Consider the group of similar medical images to be compressed. Example: MRI images.
2. Every column of the individual medical image is treated as the vector.
3. Compute the co-variance matrix  $C$  using the collected vectors.
4. Compute the significant eigen vectors of the  $C$  corresponding to the significant eigen values.
5. Represent each column of the image to be compressed as the linear combinations of significant vectors. The corresponding co-efficients are stored along with the significant eigen vectors as the compressed data.
6. Significant eigen vectors are common for all the images. The corresponding co-efficients along with the stored significant eigen vectors are used to reconstruct the image back.
7. Number of significant eigen vectors determines the quality of the reconstructed image.
8. 27 MRI images each with size  $128 \times 128$  is considered to demonstrate the KLT based compression.
9. Every image is reconstructed back with two different compression ratio and are displayed in the Fig. 2.3. One with 50 significant eigen vectors (compression ratio 1:2.56) and another with 25 significant eigen vectors (compression ratio 1:5.12).

### 4.3.2.1 kltformedimagecomp.m

```

%Medical image compression using KL-Transformation
%Each column of the image is treated as the vector.

load mri
D=double(D);
%collecting the vectors
vect=[];
for i=1:1:27
    vect=[vect D(:, :, 1,i)];
end
vect=double(vect);
%Computing the inner-product matrix
vect=double(vect);
C=cov(double(vect'));
[E,V]=eigs(C,50);
%Compression starts here
%The co-efficients to be stored (to store all the images ) is as follows.
%Compression for the first image

a=[25 50];
p=1;
for i=1:5:27
    I=double(D(:, :, i));
    subplot(6,3,p)
    p=p+1
    colormap(gray)
    imagesc(I)
    for cr=1:1:2
        COEF=[ ];
            E1=E(:,1:1:a(cr));
        for j=1:1:128
            COEF=[COEF E1'*I(:,j)];
        end
    %Reconstruction
    for k=1:1:128
        R(:,k)=E1*COEF(:,k);
    end
    subplot(6,3 ,p)
    colormap(gray)
    imagesc(R)
    p=p+1
end
end
end

```

## 4.4 Feature Extraction and Classification

To help the automated system to classify the medical images into normal and abnormal categories, the features are extracted from the medical image to train the classifier. The feature vector extracted using the wavelet transformation is described below.

**Table 4.1** Discrete Wavelet Transformation (DWT)

Row-wise	Column-wise	Type1
Low pass	Low pass	Approximation
Low pass	High pass	Horizontal
High pass	Low pass	Vertical
High pass	High pass	Diagonal

### 4.4.1 Using Discrete Wavelet Transformation

1. Discrete wavelet transformation (DWT) consists of four discrete  $1D$ -filters. They are low pass decomposition filter, high pass decomposition, corresponding low pass reconstruction filter and high pass reconstruction filter.
2. Every row of the medical image under consideration is passed through the  $1D$ -filter 1.
3. Every column of the obtained image is passed through the  $1D$ -filter 2.
4. The alternative samples are removed from the obtained image to obtain the Wavelet decomposed image.
5. Based on the combinations of the filter-1 and filter-2 selected from the low-pass and the high-pass decomposition filters, we obtain approximation, horizontal, vertical and the diagonal images of first level decomposition (refer Table 4.1).
6. The approximation of the first level decomposed image is treated as the original image and are subjected to decomposition to obtain second level decomposition (refer Fig. 4.11) and so on.
7. The decomposed values (significant) of last level approximation image and all the horizontal, vertical and diagonal values (upto to the last level decomposition) are collected as the vector and are treated as feature vector for further classification.
8. Based on the type of  $1D$  filters used, we obtain different wavelet decomposition. The *db6*  $1D$ -filters are used for illustration.
9. Corresponding recomposition filters are used for DWT based medical image compression and denoising. Decomposition filters are not used for feature extraction.

#### 4.4.1.1 feextwavelet.m

```

load mri
A=D(:, :, 1, 14);
[C, L]=wavedec2(A, 2, 'db6');
A=appcoef2(C, L, 'db6', 2);
[H2, V2, D2]=detcoef2('all', C, L, 2);
[H1, V1, D1]=detcoef2('all', C, L, 1);

figure
colormap(gray)

```

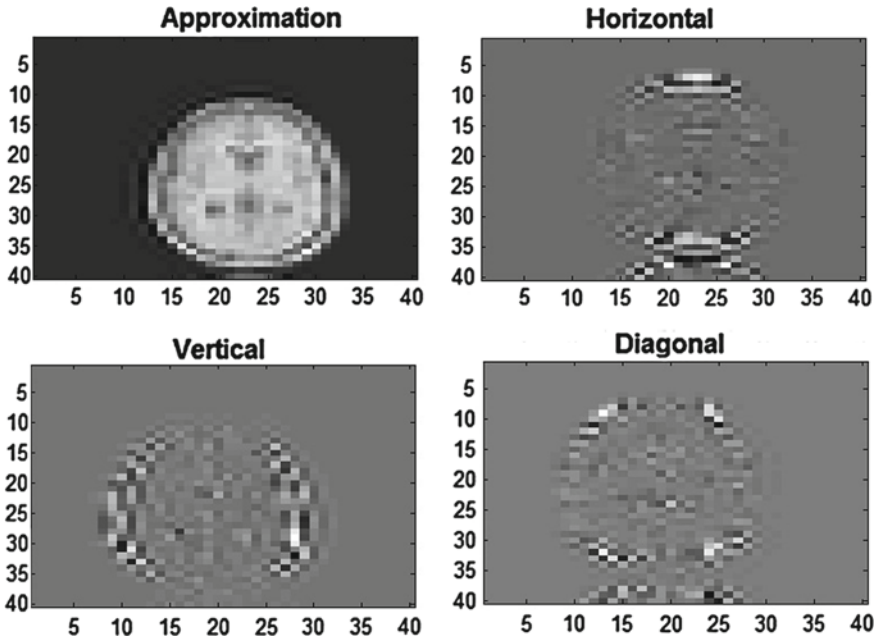


Fig. 4.11 Second level wavelet decomposed images using *db6* filter

```
subplot(2,2,1)
imagesc(A)
subplot(2,2,2)
imagesc(H2)
subplot(2,2,3)
imagesc(V2)
subplot(2,2,4)
imagesc(D2)
figure
colormap(gray)
subplot(2,2,1)
imagesc(D(:, :, 1, 14))
subplot(2,2,2)
imagesc(H1)
subplot(2,2,3)
imagesc(V1)
subplot(2,2,4)
imagesc(D1)
```

### 4.4.2 Dimensionality Reduction Using Principal Component Analysis (PCA)

The features that highlights the discrimination between the different groups are collected from the individual images and are collected as the feature vectors. The dimension of such collected vectors are usually large. There is the need to map those vectors to the lower dimensional space for further classification. This is achieved by identifying the transformation matrix  $W^T$  of size  $m \times n$  that maps the vector  $v$  of size  $n \times 1$  to the vector  $u$  of size  $m \times 1$ , where  $m \ll n$ . One way to achieve the same is to use the KL-Transformation (refer Sect. 4.3.2) and are known as Principal Component Analysis (PCA). The steps involved in dimensionality reduction using PCA, followed by the simple euclidean distance based classifier is summarized below.

1. Consider the first 24 frames of the MRI image are grouped (refer Fig. 4.12) into four categories (based on the time it occur).
2. First three columns of the images in the Fig. 4.12 are treated as the training set. The remaining data are treated as the testing images.
3. For simplicity the actual image is resized to the size of  $12 \times 12$  and are reshaped to  $1 \times 144$  to obtain the feature vector. (Note that in practice, suitable feature extraction is applied).
4. Apply KL-Transformation (refer Sect. 4.3.2) to obtain the significant eigen vectors.
5. The dimension of the projected vector depends on the selection of the number of significant eigen vectors.
6. For instance, if one eigen vector is selected, the dimension of the projected vector is  $1 \times 1$ . If two significant vectors are selected, the dimension of the projected vector is  $2 \times 1$ . The projection using the PCA basis maximizes the variances of the individual elements of the projected vectors (refer Appendix B).
7. The projected 1D and 2D vectors corresponding to the training set is displayed in the Figs. 4.13 and 4.14 respectively.
8. The mean of the projected vectors corresponding to the individual groups are computed and are stored for further classification of unknown vectors.
9. For the arbitrary test image(I), feature vector is collected and are projected to the lower dimensional space using the corresponding transformation matrix. The euclidean distance of the projected vector with the mean vectors of the individual groups of the projected vectors are computed. Identify the group that belongs to the lowest euclidean distance (say group  $G$ ) and declare that the arbitrary test image (I) belongs to the group  $G$ .
10. The classification procedure (based on eculidean distance) is repeated for all the test images and the corresponding declared group number is displayed in Table 4.3 along with the corresponding original group number. The percentage of correct classification are given as 66.67 and 50 respectively with the 1D and 2D projection.

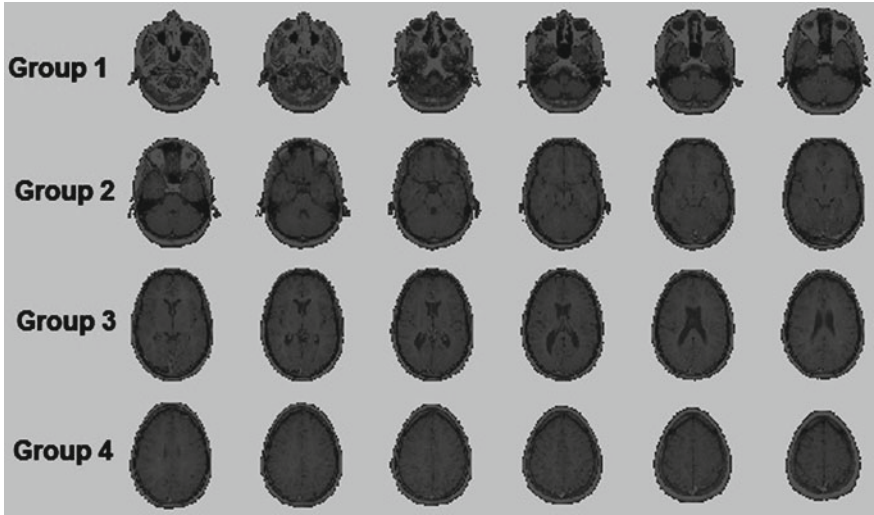


Fig. 4.12 First 24 Frames of the MRI image grouped based on the time it occurs

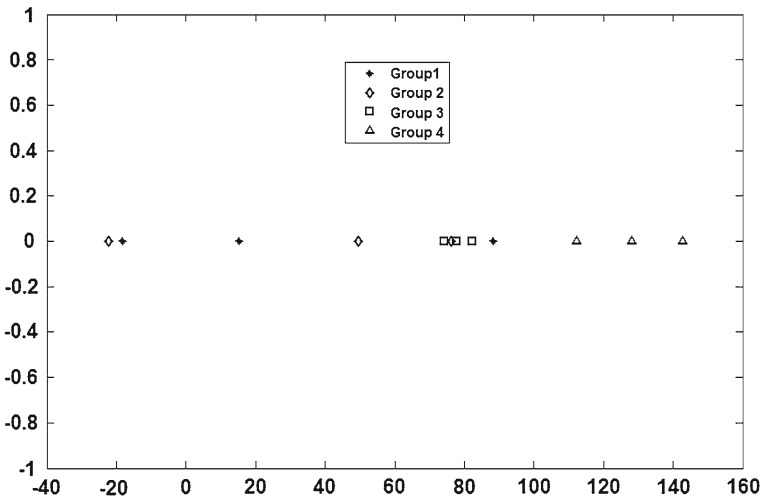


Fig. 4.13 One dimensional projection corresponding to the most significant eigen vector (PCA basis)

11. The eculidean distance based classifier is used for illustration. But there are other classifiers such as Support Vector Machine (SVM), Artificial Neura Network (ANN), Bayesian classifier , etc. are used in practice to improve the classification rate.



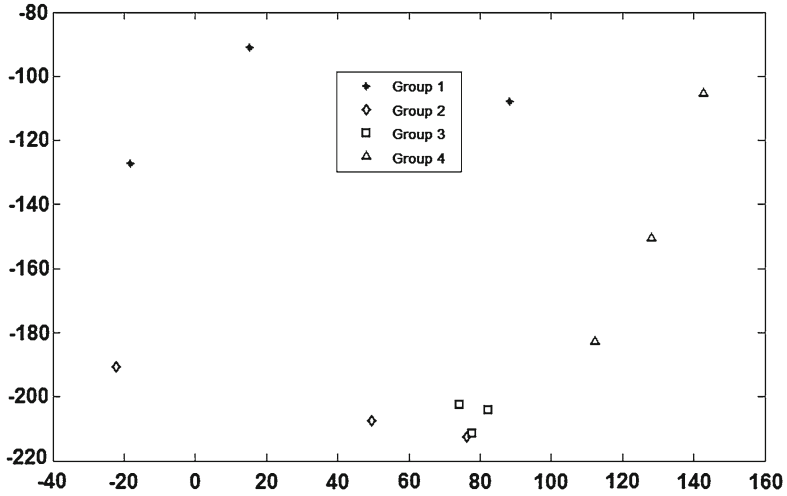


Fig. 4.14 Two dimensional projection with two eigen vectors corresponding to the largest two eigen values (PCA basis)

#### 4.4.2.1 projection1d-2d-pca.m

```

%PCA based Dimensionality reduction

load mri
figure
colormap(map)
montage(D)
tr=[1 3 5 7 9 11 13 15 17 19 21 23];
COLLECT=[ ];
D=double(D);
for i=1:1:12
    D1=imresize(D(:,:,1, tr(i)), [12 12]);
    COLLECT=[COLLECT; reshape(D1,1,12*12)];
end
C=cov(COLLECT,1);

%1D projection for the training data
[EV1,DV1]=eigs(C,1);
ONEDPV=EV1'*COLLECT'
figure
plot(ONEDPV(1:3),zeros(1,3),'*')
hold on
plot(ONEDPV(4:6),zeros(1,3),'d')
hold on
plot(ONEDPV(7:9),zeros(1,3),'s')
hold on
    
```

```

plot(ONEDPV(10:12),zeros(1,3),'^')
MEANVECTOR1=[mean(ONEDPV(1:3)) mean(ONEDPV(4:6)) ...
mean(ONEDPV(7:9)) mean(ONEDPV(10:12)) ];
%2D projection of the training ) data
figure
[EV2,DV2]=eigs(C,2);
TWODPV=EV2'*COLLECT';
MEANVECTOR2=[mean(TWODPV(:,1:3)') ;mean(TWODPV(:,4:6)') ;...
mean(TWODPV(:,7:9)') ; mean(TWODPV(:,10:12)') ] ;
plot(TWODPV(1,1:3),TWODPV(2,1:3),'*')
hold on
plot(TWODPV(1,4:6),TWODPV(2,4:6),'d')
hold on
plot(TWODPV(1,7:9),TWODPV(2,7:9),'s')
hold on
plot(TWODPV(1,10:12),TWODPV(2,10:12),'^')

tr=[2 4 6 8 10 12 14 16 18 20 22 24];
COLLECT=[];
D=double(D);
for i=1:1:12
    D1=imresize(D(:,:,1,tr(i)),[12 12]);
    COLLECT=[COLLECT; reshape(D1,1,12*12)];
end
%1D projection for the testing data
ONEDPV=EV1'*COLLECT';
DETECTEDAS1D=[];
for i=1:1:12
    [P,Q]= min(( repmat(ONEDPV(i),1,4)-MEANVECTOR1).^2);
    DETECTEDAS1D=[DETECTEDAS1D Q];
end
REF=[1 1 1 2 2 2 3 3 3 4 4 4];
POS1D=length(find((REF-DETECTEDAS1D)==0))/12;

%2D projection of the testing data
TWODPV=EV2'*COLLECT';
DETECTEDAS2D=[];
for i=1:1:12
    [P,Q]= min(sum(repmat(TWODPV(:,i),1,4)-MEANVECTOR2').^2);
    DETECTEDAS2D=[DETECTEDAS2D Q];
end
POS2D=length(find((REF-DETECTEDAS2D)==0))/12;

```

### 4.4.3 Dimensionality Reduction Using Linear Discriminant Analysis (LDA)

The dimensionality reduction using PCA does not take care of discrimination of the groups. In LDA, we try to get the transformation matrix such that after projection, the vectors that belong to the identical group comes closer to each other. At that same time, the centroid of various clusters are made apart (refer Appendix B). This is achieved as follows.

1. The within-cluster scatter matrix (refer 4.6) and the between-cluster scatter matrix (refer 4.7) are computed.
2. Compute the eigen vectors that correspond to the significant eigen values of the matrix  $S_B^{-1} S_W$ .
3. Arrange the selected eigen vectors in the column form to obtain the transformation matrix  $W_{LDA}$ .
4. The experiment described in the Sect. 4.4.2 is repeated with the matrix  $W_{LDA}$ .
5. The projected 1D and 2D vectors corresponding to the training set using LDA is displayed in the Figs. 4.15 and 4.16 respectively.
6. The procedure for euclidean distance classification (refer Sect. 4.4.2) is done for all the test images (using  $W_{LDA}$ ) and the corresponding classified group is displayed in Table 4.3 along with the corresponding original group index. The percentage of correct classification are given as 50 and 66.67 respectively with the 1D and 2D projection.

$$S_W = \sum_{i=1}^r n_i cov_i \quad (4.6)$$

$$S_B = \sum_{i=1}^r n_i (C_i - C)(C_i - C)^T \quad (4.7)$$

where  $r$  is the number of groups (clusters)  $cov_i$  is the co-variance matrix of the  $i$ th cluster,  $n_i$  is the number of vectors in the  $i$ th cluster,  $C_i$  is the centroid vector of the  $i$ th cluster and  $C$  is the mean vector of centroids of all the clusters.

#### 4.4.3.1 projection1d-2d-lda.m

```
%LDA based dimensionality reduction
load mri
figure
colormap(map)
montage(D)
tr=[1 3 5 7 9 11 13 15 17 19 21 23];
D=double(D);
SW=0;
CENTROID=[];
```

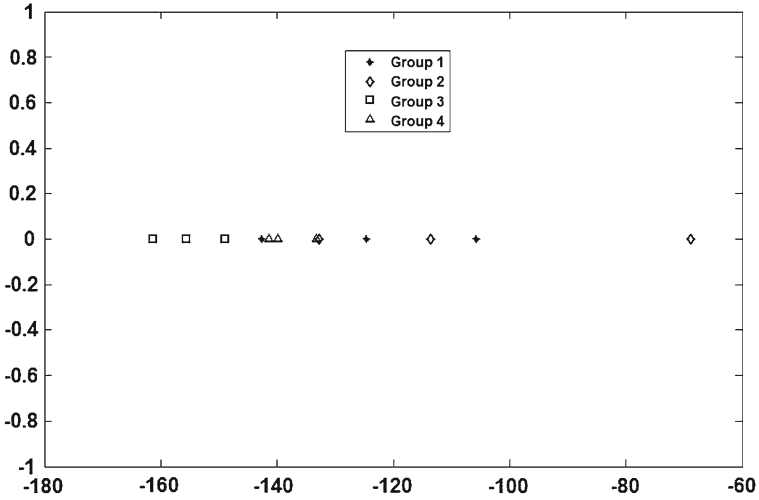


Fig. 4.15 One dimensional projection corresponding to the most significant eigen vector (LDA basis)

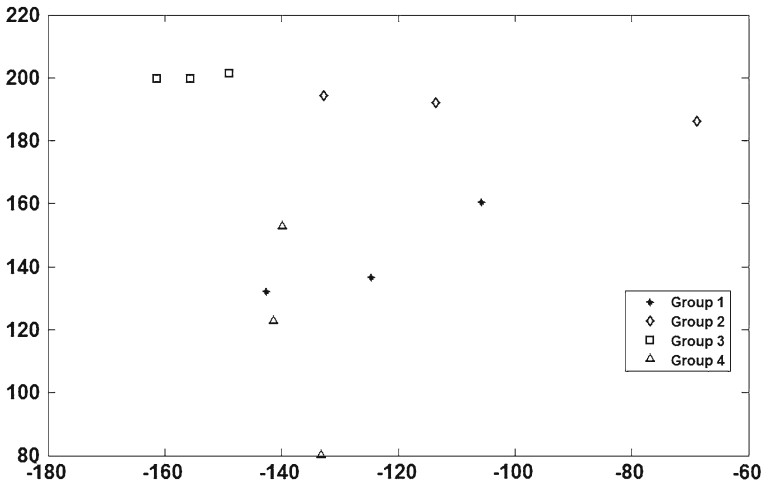


Fig. 4.16 Two dimensional projection with two eigen vectors corresponding to the largest two eigen values (LDA basis)

```
for j=1:1:4
COLLECT=[];
for i=1:1:3
D1=imresize(D(:,:,1, tr(3*(j-1)+i)), [12 12]);
COLLECT=[COLLECT; reshape(D1,1,12*12)];
end
CENTROID=[CENTROID;mean(COLLECT)];
```

```

SW=SW+cov(COLLECT);
end
SB=cov(CENTROID);

COLLECT=[ ];
for i=1:1:12
    D1=imresize(D(:,:,1, tr(i)), [12 12]);
    COLLECT=[COLLECT; reshape(D1,1,12*12)];
end
%1D projection for the training data
[E1, V1]=eigs(pinv(SB)*SW, 1);
ONEDPV=E1'*COLLECT'
figure
plot(ONEDPV(1:3), zeros(1,3), '*')
hold on
plot(ONEDPV(4:6), zeros(1,3), 'd')
hold on
plot(ONEDPV(7:9), zeros(1,3), 's')
hold on
plot(ONEDPV(10:12), zeros(1,3), '^')
MEANVECTOR1=[mean(ONEDPV(1:3)) mean(ONEDPV(4:6))...
    mean(ONEDPV(7:9)) mean(ONEDPV(10:12)) ];

%2D projection of the training data
figure
[E2, V2]=eigs(pinv(SB)*SW, 2);
TWOOPV=E2'*COLLECT';
MEANVECTOR2=[mean(TWOOPV(:,1:3)) ;mean(TWOOPV(:,4:6)) ;...
    mean(TWOOPV(:,7:9)) ; mean(TWOOPV(:,10:12)) ];
plot(TWOOPV(1,1:3), TWOOPV(2,1:3), '*')
hold on
plot(TWOOPV(1,4:6), TWOOPV(2,4:6), 'd')
hold on
plot(TWOOPV(1,7:9), TWOOPV(2,7:9), 's')
hold on
plot(TWOOPV(1,10:12), TWOOPV(2,10:12), '^')

tr=[2 4 6 8 10 12 14 16 18 20 22 24];
COLLECT=[];
D=double(D);
for i=1:1:12
    D1=imresize(D(:,:,1, tr(i)), [12 12]);
    COLLECT=[COLLECT; reshape(D1,1,12*12)];
end

%1D projection for the testing data
ONEDPV=E1'*COLLECT';
DETECTEDAS1D=[];
for i=1:1:12

```

```

[P,Q]= min(( repmat( ONEDPV(i) , 1, 4) -MEANVECTOR1) .^2);
DETECTEDAS1D=[DETECTEDAS1D Q];
end
REF=[1 1 1 2 2 2 3 3 3 4 4 4];
POS1D=length(find((REF-DETECTEDAS1D)==0))/12;

%2D projection of the testing data
TWODPV=E2'*COLLECT';
DETECTEDAS2D=[];
for i=1:1:12
[P,Q]= min(sum(repmat(TWODPV(:,i) , 1, 4) -MEANVECTOR2') .^2);
DETECTEDAS2D=[DETECTEDAS2D Q];
end
POS2D=length(find((REF-DETECTEDAS2D)==0))/12;

```

#### 4.4.4 Dimensionality Reduction Using Kernel-Linear Discriminant Analysis (K-LDA)

Although discrimination is taken care in LDA, the clusters in the projected space are overlapping in nature. Intuitively, we understand that when the vectors are mapped to the higher dimensional space, there is the chance to have more separation between the clusters. The LDA formulated using the mapped vectors in the higher dimensional space involves the inner-product of the mapped vectors rather than the mapped vector itself. (i.e) Let the vector  $u$  and  $v$  are present in the lower dimensional space. Also let the corresponding transformed vectors in the higher dimensional space be represented as  $\phi(u)$  and  $\phi(v)$  respectively. The LDA formulated using the projected vectors doesn't involve  $\phi(u)$  and  $\phi(v)$  explicitly (refer Appendix B), instead it involves  $\phi(u)^T \phi(v)$  (the inner-product values of the projected vectors). Hence if we are able to find the function  $k$  (kernel function) such that  $k(u, v) = \phi(u)^T \phi(v)$ , there is no need of explicit mapping to the higher dimensional space. This lead to the technique called Kernel-Linear discriminant analysis as described below.

1. Arrange all the training vectors in the column form to obtain the matrix  $M$ .
2. Compute the kernel matrix  $K = k(M^T M)$ , where  $k$  is the kernel function. The list of mostly used kernel function is listed in the Table 4.2.
3. The columns of the matrix  $K$  are considered as the training vectors in the inner-product space.
4. Compute the LDA basis (refer Sect. 4.4.3) in the inner-product space. Arrange them in the columnwise to obtain the transformation matrix  $W_K$  to map to the lower dimensional space.
5. Any arbitrary vector  $v$  in the feature space is mapped to the kernel space using  $k(M^T v)$ . This is further mapped to the lower dimensional space using the transformation matrix  $W_K$  as  $W_K k(M^T v)$ .

**Table 4.2** List of mostly used kernel functions

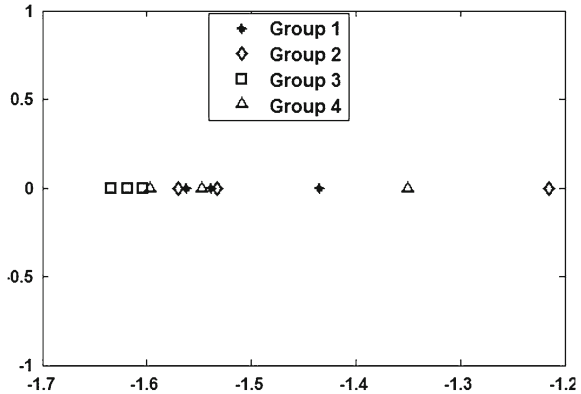
Name	Kernel function $k(x_1, x_2)$
Innerproduct kernel	$(x_1^T x_2)$
Gaussian	$\exp \frac{-\ x_1 - x_2\ ^2}{c}$
Polynomial	$(x_1^T x_2 + c)^3$

**Table 4.3** Classification using PCA,LDA and K-LDA projection

ORIGINAL GROUP	1	1	1	2	2	2	3	3	3	4	4	4
Classified using 1D PCA	3	1	1	1	3	3	3	3	3	4	4	4
Classified using 2D PCA	4	3	2	2	3	3	3	3	1	4	4	4
Classified using 1D LDA	3	2	2	2	1	4	3	3	3	4	4	1
Classified using 2D LDA	4	1	2	2	2	3	3	3	1	4	4	4
Classified using 1D K-LDA <sup>a</sup>	3	2	2	2	3	3	3	3	3	3	4	2
Classified using 2D K-LDA <sup>a</sup>	1	1	2	2	3	2	3	3	1	4	4	4

<sup>a</sup> Using gaussian kernel

**Fig. 4.17** One dimensional projection corresponding to the most significant eigen vector (K-LDA basis)

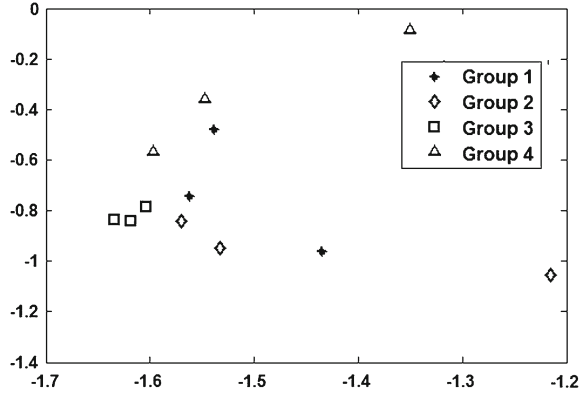


6. The experiment described in the Sect. 4.4.2 is repeated with the transformation matrix  $W_K$  and the transformation  $W_K k(M^T v)$ .
7. The projected 1D and 2D vectors corresponding to the training set using K-LDA is displayed in the Figs. 4.17 and 4.18 respectively.
8. The procedure for eculidean distance classification (refer Sect. 4.4.2) is done for all the testing images (using the transformation matrix  $W_K$  and the transformation  $W_K k(M^T v)$ ) and the corresponding classified group is displayed in Table 4.3 along with the corresponding original group index. The percentage of correct classification are given as 41.67 and 75 respectively with the 1D and 2D projection.

**4.4.4.1 projection1d-2d-klda.m**

```
%KLDA based dimensionallity reduction
load mri
```

**Fig. 4.18** Two dimensional projection with two eigen vectors corresponding to the largest two eigen values (K-LDA basis)



```

figure
colormap(map)
montage(D)
tr=[1 3 5 7 9 11 13 15 17 19 21 23];
D=double(D);
COLLECT1=[ ];
for i=1:1:12
    D1=imresize(D(:,:,1, tr(i)), [12 12]);
    COLLECT1=[COLLECT1; reshape(D1,1,12*12)];
end
COLLECT1=COLLECT1';
%Computing the kernel matrix
K=[];
for i=1:1:12
    for j=1:1:12
        K(j,i)=k11(COLLECT1(:,i),COLLECT1(:,j),1)
    end
end

SW=0;
CENTROID=[];
for j=1:1:4
    COLLECT=K(:, (3*(j-1)+1):(3*(j-1)+3));
    CENTROID=[CENTROID;mean(COLLECT')];
    SW=SW+cov(COLLECT');
end
SB=cov(CENTROID);

%1D projection for the training data
[E1,V1]=eigs(pinv(SB)*SW,1);
ONEDPV=E1'*K';
figure
plot(ONEDPV(1:3),zeros(1,3),'*')
    
```



```

hold on
plot(ONEDPV(4:6),zeros(1,3),'d')
hold on
plot(ONEDPV(7:9),zeros(1,3),'s')
hold on
plot(ONEDPV(10:12),zeros(1,3),'^')
MEANVECTOR1=[mean(ONEDPV(1:3)) mean(ONEDPV(4:6)) ...
mean(ONEDPV(7:9)) mean(ONEDPV(10:12)) ];
%2D projection of the training data
figure
[E2,V2]=eigs(pinv(SB)*SW,2);
TWODPV=E2'*K';
MEANVECTOR2=[mean(TWODPV(:,1:3)') ;mean(TWODPV(:,4:6)') ;...
mean(TWODPV(:,7:9)') ; mean(TWODPV(:,10:12)') ];
plot(TWODPV(1,1:3),TWODPV(2,1:3),'*')
hold on
plot(TWODPV(1,4:6),TWODPV(2,4:6),'d')
hold on
plot(TWODPV(1,7:9),TWODPV(2,7:9),'s')
hold on
plot(TWODPV(1,10:12),TWODPV(2,10:12),'^')

tr=[2 4 6 8 10 12 14 16 18 20 22 24];
D=double(D);
COLLECT2=[];
for i=1:1:12
    D1=imresize(D(:,:,1,tr(i)),[12 12]);
    COLLECT2=[COLLECT2; reshape(D1,1,12*12)];
end
COLLECT2=COLLECT2';
Ktest=[];
for i=1:1:12
    for j=1:1:12
        Ktest(j,i)=k11(COLLECT2(:,i),COLLECT1(:,j),1);

    end
end

%1D projection for the testing data
ONEDPV=E1'*Ktest;
DETECTEDAS1D=[];
for i=1:1:12
    [P,Q]= min(( repmat(ONEDPV(i),1,4)-MEANVECTOR1).^2);
    DETECTEDAS1D=[DETECTEDAS1D Q];
end
REF=[1 1 1 2 2 2 3 3 3 4 4 4];
POS1D=length(find((REF-DETECTEDAS1D)==0))/12;

%2D projection of the testing data

```

```
TWODPV=E2'*Ktest;
DETECTEDAS2D=[];
for i=1:1:12
[P,Q]= min(sum(repmat(TWODPV(:,i),1,4)-MEANVECTOR2').^2);
DETECTEDAS2D=[DETECTEDAS2D Q];
end
POS2D=length(find((REF-DETECTEDAS2D)==0))/12;
```

In general the percentage of classification depends on the following.

1. Type of the feature vector.
2. Type of the projection technique.
3. Dimension of the projected vectors.
4. Number of training data.
5. Type of the classifier.

# Appendix A

## Solving Bloch Equation with $A\Delta v \text{sinc}(\Delta vt)$ Envelope

Rewriting (2.21) and (2.23) with  $A\Delta v \text{sinc}(\Delta vt)$  envelope as follows with the condition  $\Delta v < \frac{B_0\gamma}{2\pi}$ .

$$\frac{dM_{xy}(t)}{dt} = j\gamma A\Delta v \text{sinc}(\Delta vt) M_0 \cos(\alpha) e^{j(-\gamma B_0 t + \theta)} \tag{A.1}$$

$$\frac{dM_z(t)}{dt} = \gamma A\Delta v \text{sinc}(\Delta vt) M_0 \sin(\alpha) E_0 \sin(\theta - \phi) \tag{A.2}$$

Since  $\text{sinc}(\Delta vt)$  is almost one for all  $t$  with  $\Delta v < \frac{B_0\gamma}{2\pi}$ , we treat  $\text{sinc}(\Delta vt)$  as the constant in both the equations and solution is obtained as follows

$$M_{xy}(t) = 2M_0 \sin(\alpha) e^{j(-\gamma B_0 t + \phi)} \tag{A.3}$$

$$M_z(t) = \gamma M_0 \sin(\alpha) A\Delta v \text{sinc}(\Delta vt) \sin(\theta - \phi) t + 2M_0 \cos(\alpha) \tag{A.4}$$

# Appendix B

## Projection Techniques

When the number of samples in the feature vector is large, the computational complexity in computing the decision boundary between the different clusters increases. Therefore, there is a requirement to map the feature vector into the lower dimensional space; thereby reducing the computational complexity. This mapping is achieved by using transformation matrix  $W^T$  obtained as described below.

### B.1 Principal Component Analysis

The covariance matrix of the vectors in the vector space is given as  $C_X = E((X - \mu_X)(X - \mu_X)^T)$ , where  $X$  is the random vector associated with the vectors in the vector space ( $V$ ),  $\mu_X$  is the mean vector of the vector space. The diagonal elements of the matrix  $C_X$  gives the variance of the individual elements of the random variable  $X$ . Let the size of the random vector  $X$  be  $m \times 1$  and the transformation vector  $W_1$  of size  $m \times 1$  is used to transform the random vector  $X$  to random variable  $Y_1$  as  $Y_1 = W_1^T X$ . Note that the random variable  $Y_1$  is of the size  $1 \times 1$ . Let us formulate the objective function to optimize the vector  $W_1$  such that the variance of the random variable  $Y_1$  is maximized, subject to the constraint that  $W_1^T W_1 = 1$ . Note the variance of the random variable  $Y$  is given as  $C_Y = W_1^T C_X W_1$ . The solution is obtained as follows. The lagrangean equation is given as

$$J = W_1^T C_X W_1 + \lambda(W_1^T W_1 - 1) \tag{B.1}$$

differentiating with respect to the  $W_1$  and equate to zero and solve for  $W_1$  (exploiting the property of the symmetry of the covariance matrix  $C_X$ ), we get  $C_X W_1 = -\lambda W_1$ . This indicate that the  $W_1$  is the eigen vector corresponding to the eigen value  $-\lambda$ . To maximize the function, we have to choose the eigen vector corresponding to the largest eigen value.

Thus in general the eigen vectors of the co-variance matrix  $C_X$  corresponding the largest  $n$  significant eigen values are arranged in columnwise to obtain the

transformation matrix  $W = [W_1 W_2 W_3 \dots W_n]$  to transform the random vector  $X$  to random vector  $Y = W^T X$  such that that the variances of the individual elements of the random vector  $Y$  is maximized. The column vectors of the matrix  $W$  thus obtained forms the Principal Component Analysis (PCA) basis of the vector space ( $V$ ).

## B.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) consists of two matrices. They are within-cluster scatter matrix ( $S_W$ ) and the between-cluster scatter matrix ( $S_B$ ).

$$S_W = \sum_{i=1}^r n_i \text{cov}_i \quad (\text{B.2})$$

$$S_B = \sum_{i=1}^r n_i (C_i - C)(C_i - C)^T \quad (\text{B.3})$$

where  $r$  is the number of groups (clusters)  $\text{cov}_i$  is the co-variance matrix of the  $i$ th cluster,  $n_i$  is the number of vectors in the  $i$ th cluster,  $C_i$  is the centroid vector of the  $i$ th cluster and  $C$  is the mean vector of centroids of all the clusters. We can assign  $r$  random vectors namely  $X_1, X_2, X_3 \dots X_r$  associated with every cluster. Also  $X_{ij}$  be the random variable associated with the  $j$ th element in the  $i$ th random vector. The diagonal elements of the matrix  $S_W$  is given as  $\sum_{i=1}^r \sum_{j=1}^{n_i} \text{var}(X_{ij})$  (gives the information how the vectors are closer to each other within the cluster).

Similarly the diagonal elements of the matrix  $S_B$  gives the summation of the squared distances of the centroid vectors  $C_i$  with the overall centroid vector  $C$ , scaled with  $n_i$  (which gives the information how the centroids are various clusters are far away from each other). Consider the transformation  $Y = W^T X$ , where  $W$  is the transformation matrix that maps the vector associated with the random vector  $X$  (in the higher dimensional space) to the vector associated with the random vector  $Y$  (in the lower dimensional space). Let the size of the transformation matrix  $W$  is given as  $m \times n$ . The within-scatter matrix and the between-scatter matrix associated with the random vector  $Y$  are given as  $W^T S_W W$  and  $W^T S_B W$  respectively. Thus the objective function is formulated (such that the vectors within the cluster after transformation comes closer to each other and the centroids of various clusters after transformation are separated well apart) as follows.

Maximize:  $\frac{\text{trace}(W^T S_B W)}{\text{trace}(W^T S_W W)}$ . Consider the first column of the transformation matrix be represented as  $W_1$ . The objective function is rewritten as  $\frac{W_1^T S_B W_1}{W_1^T S_W W_1}$ . The constraint  $W_1^T S_W W_1 = 1$  is introduced to solve the objective function.

The lagrangean equation is written as

$$J = W_1^T S_B W_1 + \lambda (W_1^T S_W W_1 - 1) \quad (\text{B.4})$$

differentiating the equation with respect to  $W_1$  (exploiting the property of symmetry property of the matrices  $S_B$  and  $S_W$  and equate to zero gives the solution as the eigen vector corresponding to the matrix  $S_W^{-1}S_B$  corresponding to the largest eigen value. The other columns of the matrix  $W$  are the eigen vectors corresponding to the next largest eigen values. Thus the eigen vectors corresponding to the significant eigen values of the matrix  $S_W^{-1}S_B$  are arranged column wise to obtain the matrix  $W$ . The column vectors thus obtained form the LDA basis of the vector space  $V$  associated with random variable  $X$ .

### B.3 Kernel-Linear Discriminant Analysis

To improve the performance of LDA by creating wide separation between the clusters in the higher dimensional space, Kernel-Linear Discriminant Analysis (K-LDA) is used. In this technique, the vectors corresponding to the feature dimensional space is mapped to the higher dimensional space. By intuition we understand that the map to the higher dimensional space introduces better separation. Suppose the transformation  $\phi(X)$  transforms the vector in the feature dimensional space to the higher dimensional space. The between-scatter matrix and within-scatter matrix computed using the the transformed vectors are represented as  $S\phi_B$  and  $S\phi_W$  respectively. The columns of the transformation matrix ( $W_\phi$ ) to map from this higher dimensional space to the lower dimensional space is given as the eigen vector  $v_\phi$  of  $S\phi_W^{-1}S\phi_B$  (i.e)  $S\phi_B v_\phi = \lambda_{\text{phi}} S\phi_W v_\phi$ . The vector  $v_\phi$  lies in the vector space spanned by all the training vectors ( $\phi(X)$ ) that are used to compute the  $S\phi_W$  and  $S\phi_B$ . Hence we can represent the vector  $v_\phi$  as following.

For some arbitray vector  $u$ .

$$v_\phi = [\phi(X_1) \phi(X_2) \dots \phi(X_N)]u \quad (\text{B.5})$$

$$\Rightarrow S\phi_B[\phi(X_1) \phi(X_2) \dots \phi(X_N)]u = \lambda_{\text{phi}} S\phi_W[\phi(X_1) \phi(X_2) \dots \phi(X_N)]u \quad (\text{B.6})$$

Multiplying  $[\phi(X_1) \phi(X_2) \dots \phi(X_N)]^T$  on both sides of (B.6) and representing  $SK_B = [\phi(X_1) \phi(X_2) \dots \phi(X_N)]^T S\phi_B [\phi(X_1) \phi(X_2) \dots \phi(X_N)]$  and  $SK_W = [\phi(X_1) \phi(X_2) \dots \phi(X_N)]^T S\phi_W [\phi(X_1) \phi(X_2) \dots \phi(X_N)]$ , we get the following.

$$SK_B u = \lambda_{\text{phi}} SK_W u \Rightarrow \quad (\text{B.7})$$

It can be easily verified that  $SK_B$  and  $SK_W$  are the between-scatter matrix and the within-scatter matrix respectively, computed using the corresponding columns of the Gram-matrix  $\phi(X)^T \phi(X)$ , where  $\phi(X) = [\phi(X_1) \phi(X_2) \dots \phi(X_N)]$ . Let the eigen vectors of the matrix  $SK_W^{-1}SK_B$  be arranged in the column-wise to obtain the matrix  $M$ . Then the transformation matrix that transforms the vector in the higher dimensional space ( $\phi(\cdot)$ ) to the lower dimensional space is given as  $[\phi(X_1) \phi(X_2) \dots \phi(X_N)]^T M$ . Let the arbitrary vector in the higher dimensional

space be represented as  $\phi(q)$  corresponding the feature dimensional vector  $q$ . The corresponding transformed vector in the lower dimensional space is given as

$$[\phi(X_1) \phi(X_2) \dots \phi(X_N)]^T M^T \phi(q) \Rightarrow M^T [\phi(X_1) \phi(X_2) \dots \phi(X_N)]^T \phi(q) \quad (\text{B.8})$$

$[\phi(X_1) \phi(X_2) \dots \phi(X_N)]^T \phi(q)$  is the vector in the kernel space corresponding to the vector in the feature space  $\phi(q)$ . Hence to map the vector to the lower dimensional space, the following steps are followed.

1. For the arbitrary vector  $q$  in the feature space, compute the corresponding vector in the kernel space as  $q_k = [k(X_1, q) k(X_2, q) \dots k(X_N, q)]^T$ .
2. Thus the transformed vector to the lower dimensional space is given as  $M^T q_k$ .

# Appendix C

## Hankel Transformation

Hankel transformation is applicable for the image that is circular symmetric. Hence assume  $f(x, y)$  is circular symmetric. The 2D-Fourier transformation in polar form is given as follows.

$$F(U, V) = \int_{-\pi}^{\pi} \int_0^{\infty} g(r, \theta) e^{-j2\pi r l} r dr d\theta \tag{C.1}$$

where  $l = U\cos(\theta) + V\sin(\theta)$  Note that  $g(r, \theta)$  is the polar representation of the original image  $f(x, y)$  (It is also the inverse fourier transformation of the radon transformation of  $F(U, V)$ ), where  $U = l\cos(\theta)$  and  $V = l\sin(\theta)$ . As  $f(x, y)$  is circular symmetric, the corresponding radon transformation is identical and hence  $g(r, \theta)$  is identical for all  $\theta$ . This implies that the radon transformation of  $F(U, V)$  is independent of  $\theta$ . (i.e)  $F(U, V)$  is circular symmetric and hence it is enough to compute the values for the co-ordinates on the y-axis. (i.e)  $U = 0$  and  $V = 0$  to  $\infty$ . Particularly for  $U = 0$  and  $V = q$ , we get  $l = q \sin(\theta)$ . Substituting back to the Eq. (C.1), we get

$$G(q) = \int_{-\pi}^{\pi} \int_0^{\infty} g(r) e^{-j2\pi r q \sin(\theta)} r dr d\theta \tag{C.2}$$

$$\Rightarrow G(q) = \int_{-\pi}^0 \int_0^{\infty} g(r) e^{-j2\pi r q \sin(\theta)} r dr d\theta + \int_0^{\pi} \int_0^{\infty} g(r) e^{-j2\pi r q \sin(\theta)} r dr d\theta \tag{C.3}$$

$$\Rightarrow G(q) = 2 \int_0^{\pi} \int_0^{\infty} g(r) \cos(2\pi r q \sin(\theta)) r dr d\theta \tag{C.4}$$

$G(q)$  thus obtained is the hankel transformation of  $g(r)$ .



## Appendix D

### List of m-Files

- 1.2.2.1 parallelbeamprojection.m
- 1.3.3.1 Fanbeamprojection.m
- 2.3.2.1 resmagneticmoment.m
- 3.1.1.1 protondensity.m
- 3.2.1.1 t2.m
- 3.2.1.2 relaxation.m
- 3.2.1.3 dephasing.m
- 3.2.2.1 spinechodemonstration
- 3.3.1.3 spinechopolar.m
- 3.4.0.4 t1.m
- 4.2.4.2 hankeltransformation.m
- 4.2.6.1 histogramequalization.m
- 4.3.1.1 imagecompusingdct.m
- 4.3.1.2 dctt.m
- 4.3.1.3 idctt.m
- 4.3.2.1 kltformedimagecomp.m
- 4.4.1.1 feaextwavelet.m
- 4.4.2.1 projection1d-2d-pca.m
- 4.4.2.1 projection1d-2d-lda.m
- 4.4.2.1 projection1d-2d-klda.m

# Index

T1 MRI image, 35  
T2 MRI, 35  
 $\alpha$  value, 29  
2D-DFT, 42  
2D-Fourier transformation, 3

**A**  
Approximation, 87

**B**  
Back-projection, 64  
Between-cluster, 93  
Bloch equations, 27

**C**  
Cartesian scanning, 44  
Classification, 86  
Co-variance matrix, 85  
Compression ratio, 85  
Computed tomography (CT), 1, 73  
Cumulative distribution, 80

**D**  
Dephasing, 49  
Diagonal, 87  
Dimensionality reduction, 89  
Discrete cosine transformation, 82  
Discrete wavelet transformation, 87  
Discretization, 50, 54, 69

**E**  
Euclidean distance, 93  
Eigen vectors, 85  
Envelope, 37  
Euclidean distance, 89

**F**  
Fan-beam projection, 1  
Feature extraction, 86  
Fourier transformation, 36  
Free induction decay, 35  
Frequency domain, 41

**G**  
Gradient echo, 43, 50

**H**  
Hankel transformation, 76  
Helical movement, 33  
High pass decomposition, 87  
High pass reconstruction filter, 87  
Histogram equalization, 80  
Histogram specification, 81  
Horizontal, 87

**I**  
IFFT2, 51  
Image compression, 82  
Image enhancement, 74  
Image subtraction, 74  
Interpolation, 65

**J**

Jacobian, 3

**K**

K-space, 43

Kernel-linear discriminant analysis, 96

KI-transformation, 89

**L**

Larmor frequency, 29

Linear discriminant analysis, 93

Linear filterering, 76

Linear phase delay, 49

Logarithmic display, 74

Longitudinal, 35

Low pass decomposition, 87

Low pass filter, 40

Low pass reconstruction filter, 87

**M**

Magnetic resonance imaging, 27, 49

Medical image processing, 73

**N**

Non-linear filtering, 74

Nuclear medicine image, 73

**P**

Parallel beam projection, 1

Phase component, 37

Pixel, 49

Polar scanned, 65

Polar scanning, 45

Principal component analysis, 89

Proton-density, 35, 43

**R**

Radon transformation, 1

Receiver antenna, 39

Receiver coil, 40

Refocussing, 37

Relaxation, 38

Rephasing, 61

Resolution, 65

RF, 35

**S**

Selection gradient, 45

Sinc, 36

Spatial domain, 49

Spatial matrix, 61

Spin echo, 45

Spin-spin, 38

Spiral trajectory, 33

**T**

Transformation matrix, 89

Transverse component, 32

**U**

Ultrasound image, 73

**V**

Vertical, 87

**W**

Within-cluster, 93

**X**

X-ray image, 73

**Z**

Z-component, 33

Z-gradient, 37