SAR IMAGING TUTORIAL

PREPARED BY:
MUHAMMET EMIN YANIK
PROF. MURAT TORLAK

CONTENTS

1. INTRODUCTION

In scope of the project, a two-dimensional (2-D) near-field imaging system based on the combination of synthetic aperture radar (SAR) processing techniques and the low-cost system-on-chip millimeter-wave frequency-modulated continuous-wave (FMCW) radars is designed. To create a synthetic aperture over the target scene, a two-axis automatic rail system is built and integrated with Texas Instruments IWR1443 77-GHz millimeter-wave FMCW radar sensor.

In this tutorial, simplified signal processing techniques for near-field 2-D image formation is introduced and the specifications of recorded SAR data samples are detailed.

2. SYSTEM CONFIGURATION AND RECORDED DATA FORMAT

2.1. System Configuration

To reconstruct the 2-D image of the scene, data collection is performed by moving the radar along a trajectory in $x - y$ plane, which creates a rectangular grid of measurement points, as shown in Figure 1. It is assumed that transmitting and receiving antenna are located very close to each other, then they are represented by the mid-point between them. In the established (x, y, z) Cartesian coordinate system, $x -$axis, $y -$ axis, and $z -$ axis denote horizontal, vertical, and range directions, respectively.

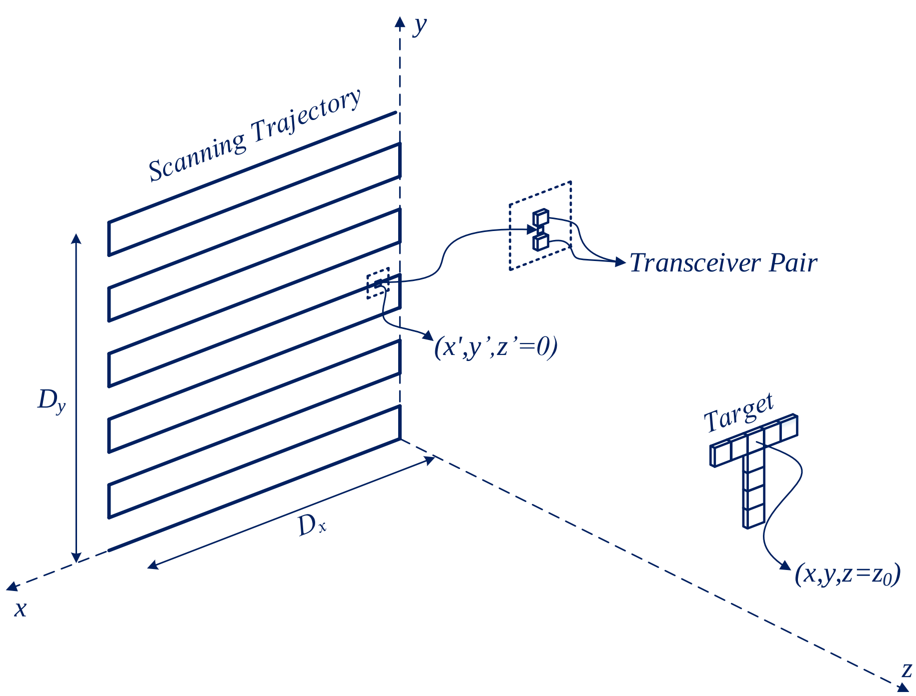

Figure 1 System Configuration

As illustrated in the measurement configuration, transceiver is at position $(x', y', 0)$ for a specific measurement instant and a general point on the target placed at the distance z_0 from the imaging system is at position (x, y, z_0). The detailed scanning aperture configuration is shown in Figure 2. Total scan size is D_x and D_y in x −axis and y − axis, respectively, and the corresponding sampling distance is d_x and d_y.

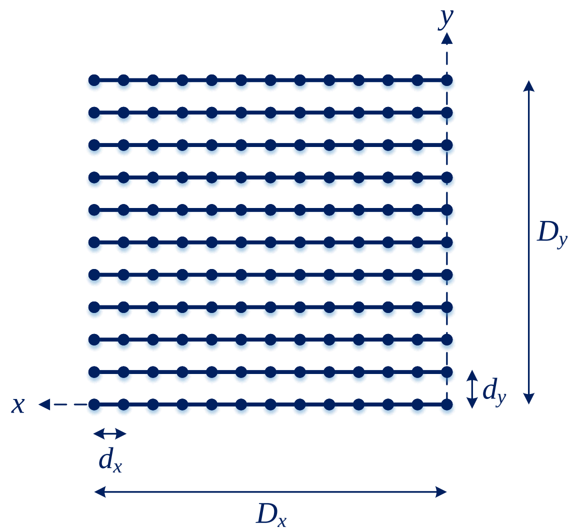

Figure 2 Aperture Configuration

Chirp paramaters used at the experiments are shown in Figure 3.

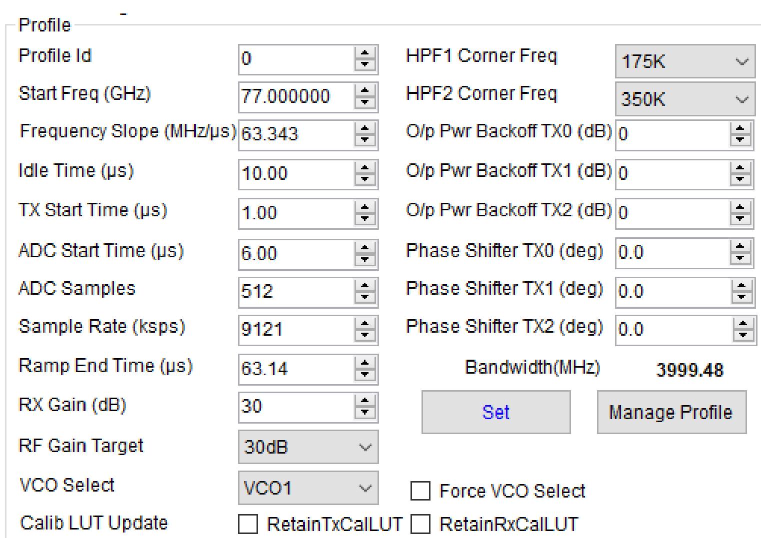

Figure 3 Chirp Parameters

2.2. Recorded Data Format

The 3-D recorded data cube is shown in Figure 4 and properties are detailed below.

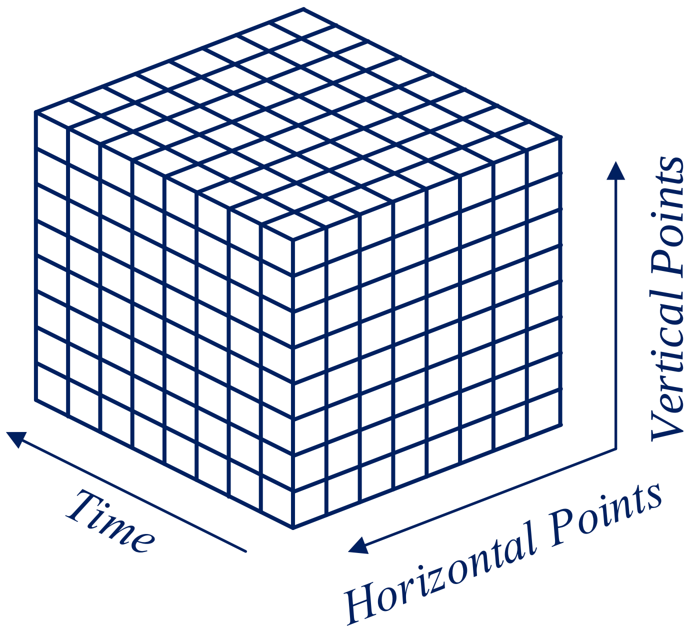

Figure 4 Recorded Data Cube

Data

rawData3D is an nSample × nVertical × nHorizontal 3-D data matrix. Details of parameters are given below.

Dimensions

nSample: Number of samples in time domain.

nVertical: Number of sample points in vertical (y) axis.

nHorizontal: Number of sample points in horizontal (x) axis.

D_x: Aperture size of horizontal (x) axis.

D_y: Aperture size of vertical (y) axis.

d_x: Sampling distance of horizontal (x) axis.

d_y: Sampling distance of vertical (y) axis.

z_0: Target distance.

3. SIMPLIFIED 2-D IMAGING ALGORITHM

After coinciding the target and aperture coordinates, the 2-D reflectivity image can be formulated as

$$f(x,y) = \mathrm{FT}_{2D}^{-1}\left[\mathrm{FT}_{2D}[s(x,y)]\mathrm{FT}_{2D}[h(x,y)]\right]$$

where FT_{2D} and FT_{2D}^{-1} denote 2-D Fourier and inverse Fourier transform operations over the $x-y$ plane, $f(x,y)$ is 2-D target reflectivity function, $s(x,y)$ is the measured radar signals, and $h(x,y)$ is the impulse response or the point spread function of the imaging system calculated for each (x,y) measurement point as

$$h(x,y) = e^{-j2k\sqrt{x^2+y^2+z_0^2}}$$

The reconstruction algorithm summarized in Figure 5 does not consider any visibility condition. Thus, built-in MATLAB 2-D FFT function can be used directly.

Algorithm Image reconstruction using matched filter approach

1: gather uniformly sampled complex FMCW data cube, $r(x,y,t)$ from transceiver over a 2-D planar aperture
2: perform range focusing to z_0 distance and gather uniformly sampled complex data, $s(x,y)$
3: perform the 2-D FFT of $s(x,y)$ to obtain sampled version of $S(k_x,k_y) = \mathrm{FT}_{2D}[s(x,y)]$
4: create matched filter, $h(x,y) = e^{-j2k\sqrt{x^2+y^2+z_0^2}}$
5: perform the 2-D FFT of $h(x,y)$ to obtain sampled version of $H(k_x,k_y) = \mathrm{FT}_{2D}[h(x,y)]$
6: multiply $S(k_x,k_y)$ and $H(k_x,k_y)$
7: perform the 2-D IFFT
8: compute the magnitude and display the data

Figure 5 Simplified Image Reconstruction Algorithm

4. RECORDED DATA SCENARIOS

4.1. Flat 2-D Target

The scenario is shown in Figure 6 and properties are detailed below.

Figure 6 Flat 2-D Target Scenario

Data Name

rawData3D_simple2D

Parameters

nSample:	512
nVertical:	100
nHorizontal:	407
$\mathbf{D_x}$:	200 mm ($\mathbf{d_x}$: 200/406 mm)
$\mathbf{D_y}$:	198 mm ($\mathbf{d_y}$: 2 mm)
$\mathbf{z_0}$:	280 mm

4.2. Two Flat 2-D Targets Concealed in Box

The scenario is shown in Figure 7 and properties are detailed below.

Figure 7 Two Flat 2-D Targets Concealed in Box Scenario

Data Name

rawData3D_twoConcealed2D

Parameters

nSample: 512

nVertical: 101

nHorizontal: 407

D_x: 200 mm (**d_x**: 200/406 mm)

D_y: 200 mm (**d_y**: 2 mm)

z_0: 260 mm (front side of the box), 320 mm (back side of the box)

4.3. Cascaded Concealed Targets

The scenario is shown in Figure 8 and properties are detailed below.

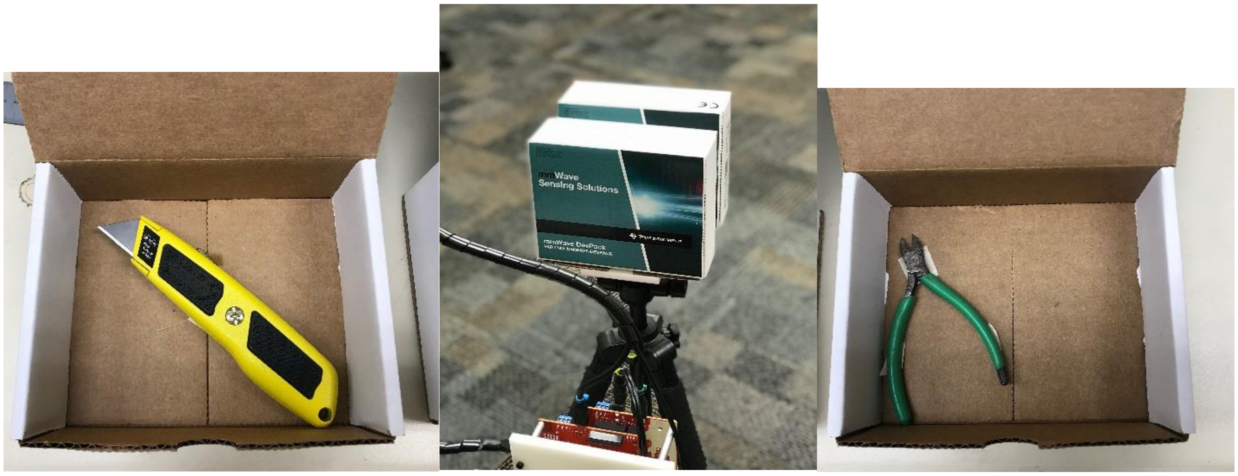

Figure 8 Cascaded Concealed Targets Scenario

Data Name

rawData3D_cascadedConcealed

Parameters

nSample: 512

nVertical: 101

nHorizontal: 407

D_x: 200 mm (**d_x**: 200/406 mm)

D_y: 200 mm (**d_y**: 2 mm)

z_0: 250 mm (first target), 340 mm (second target)

5. PROCESSING SOFTWARE

Main scripts of the processing software are identified in Table 1.

No	Script Name	Called Script
1	mainSAR.m **Definition:** Main script that preprocess the data for 2-D imaging and calls the related sub-scripts for SAR imaging.	**2, 3**
2	createMatchedFilterSimplified.m **Definition:** Script that creates matched filter.	**-**
3	reconstructSARimageMatchedFilterSimplified.m **Definition:** Script that creates the 2-D SAR image	**-**

Table 1 Software Scripts

Main Code

This code block preprocess the rawData3D for 2-D imaging and calls other SAR imaging functions.

Following parameters should be updated based on the scenario.

```
%% Load rawData3D
dataName = 'rawData3D_simple2D'; % Change only this line
rawData = load(dataName);
rawData = rawData.(dataName);

%% Define parameters, update based on the scenario
nFFTtime = 1024;     % Number of FFT points for Range-FFT
z0 = 280e-3;         % Range of target (range of corresponding image slice)
dx = 200/406;        % Sampling distance at x (horizontal) axis in mm
dy = 2;              % Sampling distance at y (vertical) axis in mm
nFFTspace = 1024;    % Number of FFT points for Spatial-FFT
```

Following line calls matched filter creation function.

```
%% Create Matched Filter
matchedFilter =
createMatchedFilterSimplified(nFFTspace,dx,nFFTspace,dy,z0*1e3);
```

Following line calls image reconstruction function. **imSize** parameter can be changed.

```
%% Create SAR Image
imSize = 200; % Size of image area in mm
sarImage =
reconstructSARimageMatchedFilterSimplified(sarData,matchedFilter,dx,dy,imSize
);
```

Matched Filter Function

matchedFilter = createMatchedFilterSimplified(xPointM,xStepM,yPointM,yStepM,zTarget)
This function creates 2-D matched filter. Input parameters and output are detailed below:

Inputs

xPointM: number of measurement points at x (horizontal) axis, should be greater than nHorizontal

xStepM: Sampling distance at x (horizontal) axis in mm

yPointM: number of measurement points at y (vertical) axis, should be greater than nVertical

yStepM: Sampling distance at y (vertical) axis in mm

zTarget: z distance of target in mm

Output

matchedFilter: 2-D yPointM × xPointM matched filter

SAR Imaging Function

sarImage = reconstructSARimageMatchedFilterSimplified(sarData,matchedFilter,xStepM,yStepM,xySizeT)
This function creates 2-D SAR image. Input parameters and output are detailed below:

Inputs

sarData: nVertical x nHorizontal 2-D SAR Data

matchedFilter: yPointM × xPointM 2-D matched filter

xStepM: Sampling distance at x (horizontal) axis in mm

yStepM: Sampling distance at y (vertical) axis in mm

xySizeT: z distance of target in mm

Output

sarImage: 2-D SAR Image

6. 2-D IMAGING RESULTS

2-D imaging result of Flat 2-D Target Scenario is given in Figure 9.

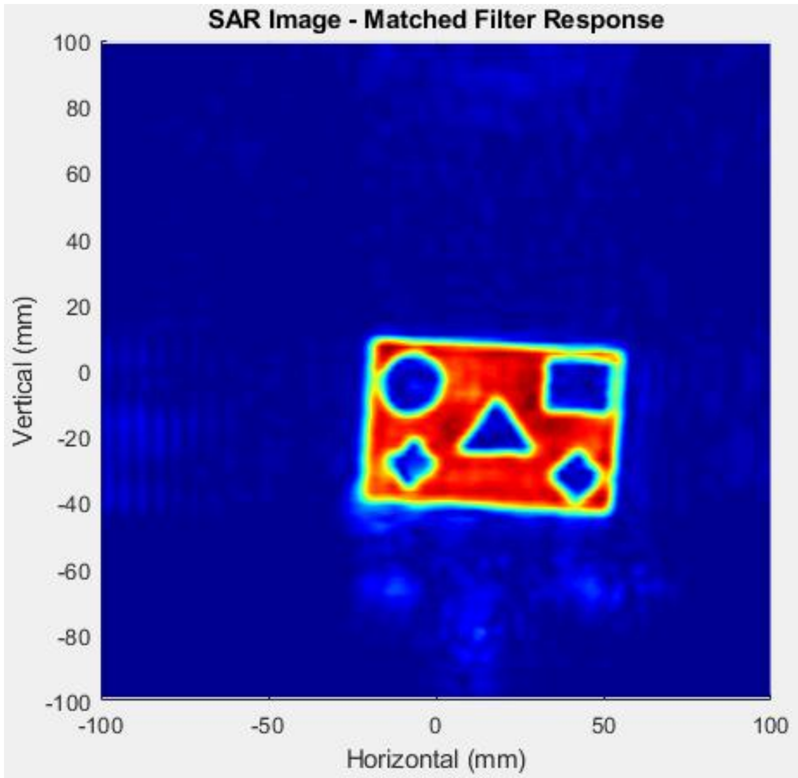

Figure 9 Flat 2-D Target Scenario Image Result

2-D imaging results of Two Flat 2-D Targets Concealed in Box Scenario are given in Figure 10. Two different slices are obtained by using two different range bins.

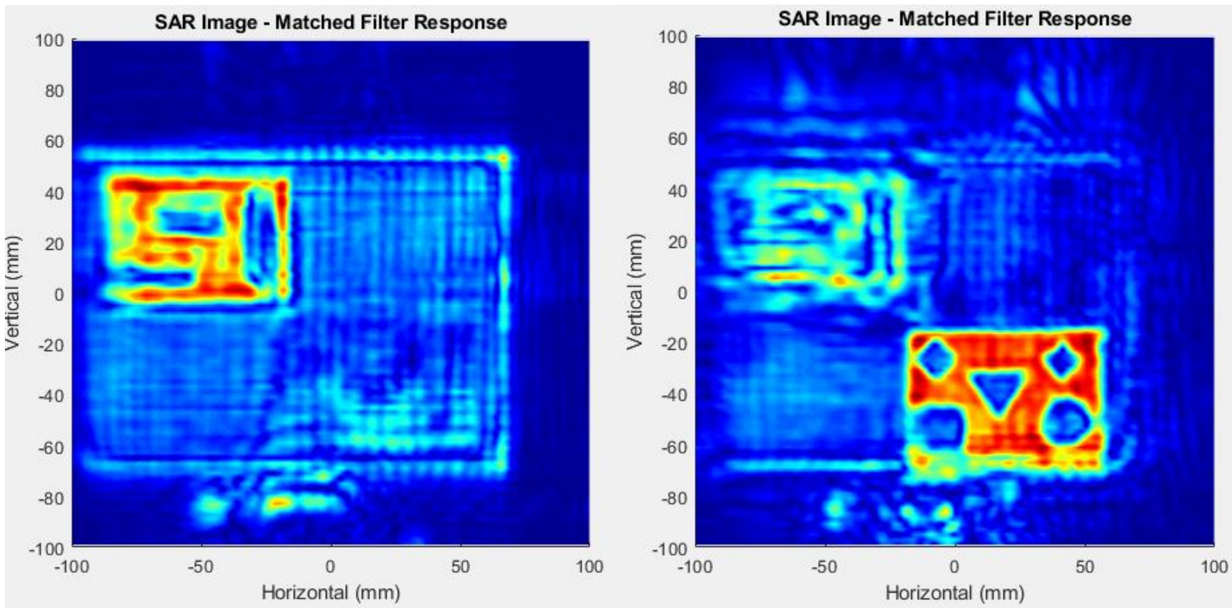

Figure 10 Two Flat 2-D Targets Concealed in Box Scenario Image Results (Note: These are initial results, will be tuned after 3-D imaging)

2-D imaging results of Cascaded Concealed Targets Scenario are given in Figure 11. Two different slices are obtained by using two different range bins.

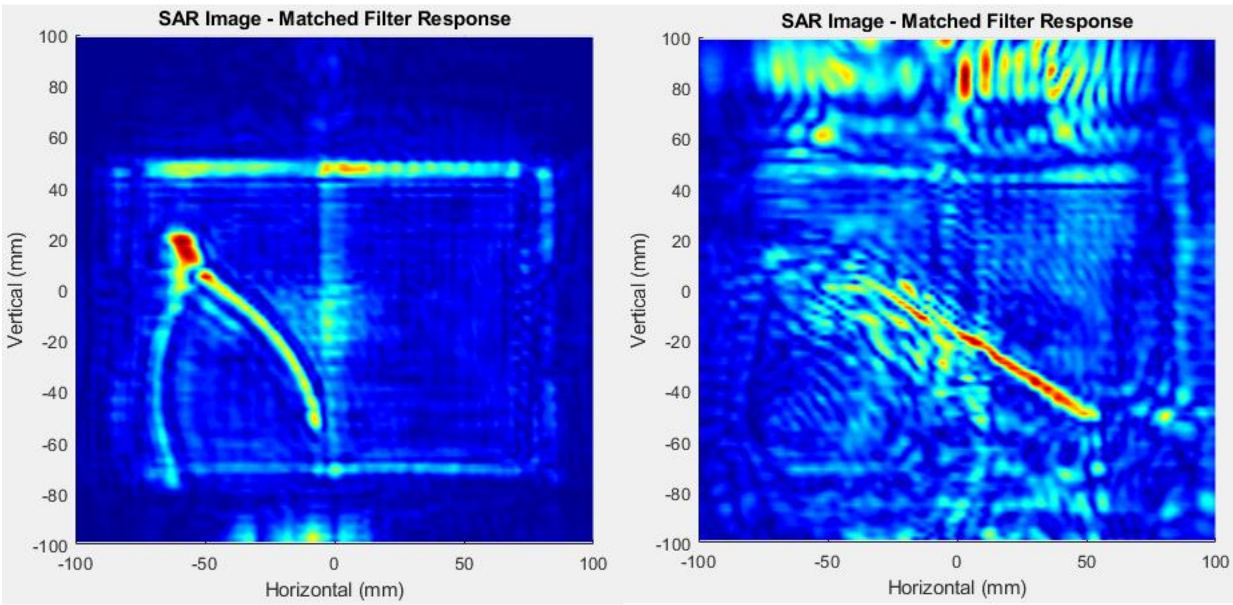

Figure 11 Cascaded Concealed Targets Scenario Image Results (Note: These are initial results, will be tuned after 3-D imaging)