

IMAGE RECONSTRUCTION BASED ON ACTIVE SCAN TECHNIQUES
IN THE TERAHERTZ FREQUENCY RANGE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

VEDAT ALI ÖZKAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
PHYSICS

JANUARY 2014

Approval of the thesis:

**IMAGE RECONSTRUCTION BASED ON ACTIVE SCAN
TECHNIQUES IN THE TERAHERTZ FREQUENCY RANGE**

submitted by **VEDAT ALI ÖZKAN** in partial fulfillment of the requirements
for the degree of **Master of Science in Physics Department, Middle East
Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Mehmet T. Zeyrek _____
Head of Department, **Physics**

Assoc. Prof. Dr. Hakan Altan _____
Supervisor, **Physics Department, METU**

Assist. Prof. Dr. Özgür Özdemir _____
Co-supervisor, **Electronics and Com. Eng. Dep., Istanbul
Technical Univ.**

Examining Committee Members:

Assoc. Prof. Dr. Hakan Altan _____
Physics Department, METU

Assist. Prof. Dr. Asaf Behzat Şahin _____
Electrical and Com. Eng. Dep., Yıldırım Beyazıt University

Assist. Prof. Dr. Özgür Özdemir _____
Electronics and Com. Eng. Dep., Istanbul Technical Univ.

Prof. Dr. Mehmet Parlak _____
Physics Department, METU

Assoc. Prof. Dr. Serhat Çakır _____
Physics Department, METU

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: VEDAT ALI ÖZKAN

Signature :

ABSTRACT

IMAGE RECONSTRUCTION BASED ON ACTIVE SCAN TECHNIQUES IN THE TERAHERTZ FREQUENCY RANGE

Özkan, Vedat Ali

M.S., Department of Physics

Supervisor : Assoc. Prof. Dr. Hakan Altan

Co-Supervisor : Assist. Prof. Dr. Özgür Özdemir

January 2014, 114 pages

THz continuous wave (CW) imaging systems have attracted interest in the past decade since they have the ability to detect non-metal threats such as ceramic knives. Moreover due to low energy levels of THz signals, these systems are not harmful. Although various types of imaging systems have been developed, the nature of THz waves has prevented the scientific community from producing a fast, high resolution and cost effective imaging system. In this study we have investigated both experimentally and theoretically the application THz waves in various imaging system configurations. The main goal was to understand the effects of the optical system design as well as noise and other parameters that could impact the formation of images in an active THz imager based on frequencies near 340GHz. The designed optical system was simulated and the effects of scanning the target field were assessed in the simulated images of various targets. The results show that reliable target detection will be aided by post-processing the images using various techniques. Furthermore, to experimentally demonstrate an active imaging system, compressive sensing was used

in a non-scanning reflection based optical geometry. In these experiments a mm-wave source near 120GHz was used to obtain the images. The obtained images were analysed and compared to that of simulations. The results show that compressive sensing methods can be implemented successfully to improve image acquisition in typically low-resolution THz imaging systems. Finally, the effect of imaging in the terahertz frequency range was also investigated for passive imaging methods. A commercial system working in the mm-wave range was used to obtain raw images at stand-off distances. The goal was to process the images without using the routines supplied by the manufacturer. Using well-established routines these raw images were processed successfully. Throughout this study careful attention is paid on the limitations on speed and resolution of these imaging systems and discussions on the applicability of the methods developed within this thesis are discussed throughout.

Keywords: Terahertz, Imaging , Compressive Sensing, Image Processing, Point Spread Function

ÖZ

AKTİF TARAMA TEKNİKLERİ KULLANARAK TERAHERTZ FREKANS ARALIĞINDA GÖRÜNTÜ OLUŞTURMA

Özkan, Vedat Ali

Yüksek Lisans, Fizik Bölümü

Tez Yöneticisi : Doç. Dr. Hakan Altan

Ortak Tez Yöneticisi : Yrd. Doç. Dr. Özgür Özdemir

Ocak 2014 , 114 sayfa

Terahertz (THz) sürekli dalga görüntüleme sistemleri geçen yıllar içinde ilgi görmeye başladı. Bu sistemlerin metal olmayan tehditleri saptayabilme yetenekleri, ilgi çekmelerinde etkili oldu. Ayrıca düşük enerji seviyelerinde çalışıyor oldukları için bu sistemler ayrıca canlılar için zararlı da değil. Birçok farklı görüntüleme sistemi geliştirilmiş olmasına rağmen THz dalgaların doğası gereği hızlı, yüksek çözünürlüklü ve ucuz bir sistemin geliştirilmesi henüz mümkün olmadı. Bu çalışmada THz dalgalarının gerek teorik gerekse deneysel olarak birçok görüntüleme biçiminde kullanımı incelendi. Bu çalışmanın esas amacı, 340 GHz civarı frekanslarda çalışan bir THz görüntüleyicinin performansına, kullanılan optik tasarım, gürültü seviyesi ve diğer parametrelerin etkilerini incelemektir. Optik tasarım bu amaçla bilgisayarda modellendi ve görüntüleme alanı modellemesi de farklı görüntüler kullanarak gerçekleştirildi. Sonuçlar gösterdi ki çeşitli algoritmaların kullanılması güvenilir bir hedef saptaması için yardımcı olacaktır.

Aktif bir görüntüleme sistemini deneysel olarak gerçekleştirmek için sıkıştırılmalı algılama prensibine uygun bir görüntüleme sistemi ile görüntüleme yapıldı. Bu sistem taramaya dayanmayan, yansımaya dayalı bir prensiple çalıştı. Bu deneylerde 120 GHz civarı frekanslarda çalışan bir milimetre dalga kaynağı kullanıldı. Deneylerden elde edilen görüntülerle bilgisayar modellemeleri kıyaslandı. Sonuçlar gösterdi ki, sıkıştırılmalı algılama prensibine uygun görüntüleme sistemleri ile normalde düşük çözünürlüklü görüntüleme sistemleri başarılı bir şekilde geliştirilebiliyor.

Son olarak, THz frekansları kullanan pasif görüntüleme metotları incelendi. Milimetre dalgaları kullanan bir ticari görüntüleme sistemi sayesinde, görüntüleme uzaklığında işlenmemiş görüntüler elde edildi. Amaç bu görüntüleri üreticinin sağladığı programları kullanmadan işlemektir. Köklü programlar kullanarak bu işlenmemiş görüntüler başarılı bir şekilde işlendi.

Bu çalışma boyunca bahsedilen görüntüleme sistemlerinin görüntüleme hızları ve çözünürlükleri yakından incelendi. Ayrıca tez içinde geliştirilen metotların uygulanabilirliği tez boyunca tartışıldı.

Anahtar Kelimeler: Terahertz, Görüntüleme, Sıkıştırılmalı Algılama, Görüntü İşleme, Nokta Yayılma Fonksiyonu

To my dear sister and her little baby

ACKNOWLEDGMENTS

Sincerest thanks to my thesis advisor Hakan Altan who not only helped me through my academic problems but did his best to keep our research team motivated with various methods. I will always miss the group dinners and visits to Up-town. I would like to also thank our research group for all their support with experimental setups and their friendship throughout these years. They were not only co-workers but also friends to me. Special thanks to all my friends who helped me keep my sanity in most stressful times. In particular I would like to thank Taylan Takan and Hakan Keskin for their software support for this thesis.

Words are not enough to express my thanks to financial supporters. Throughout my 11 semesters in METU I was supported by TEV and TUBITAK. Without them I would never be able to complete my education successfully.

I'd like to thank Republic of Turkey Ministry of Science, Industry and Technology and ASELSAN for supporting this work through San-Tez Project Number 0994STZ2011-2.

Finally I would like to express my gratitude for my parents who always supported my decisions and loved me unconditionally.

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS	xi
LIST OF TABLES	xiv
LIST OF FIGURES	xv
LIST OF ABBREVIATIONS	xviii
CHAPTERS	
1 INTRODUCTION	1
1.1 TERAHERTZ IMAGING	2
1.2 IMAGE RECONSTRUCTION	4
1.3 IMAGE PROCESSING	5
1.4 COMPRESSIVE SENSING	7
2 THEORETICAL BACKGROUND OF THZ IMAGING	11
2.1 SOURCE	11
2.2 OPTICAL INSTRUMENTS	12
2.2.1 Point Spread Function	14

2.2.2	Spatial Resolution	15
2.3	TARGET	16
2.4	RECEIVER	18
2.5	NOISE	18
2.6	ENCODING	19
2.6.1	CS Imaging System	19
2.6.2	CS Encoding	20
2.7	DECODING	21
3	THEORETICAL BACKGROUND OF ACTIVE IMAGING SYSTEM MODELING	25
3.1	PSF	26
3.1.1	Modeling PSF of a single optical element	26
3.1.2	Modeling PSF of an optical system	27
3.2	Noise	28
3.2.1	Modeling noises from environment	29
3.2.2	Modeling noises from receiver	29
4	IMAGING SYSTEMS	31
4.1	340 GHz ACTIVE IMAGING SYSTEM	31
4.2	COMPRESSIVE SENSING IMAGING SYSTEM	36
4.3	THz PASSIVE IMAGING SYSTEM	40
5	RESULTS	43
5.1	CS IMAGING SYSTEM SIMULATIONS	44
5.2	CS IMAGING SYSTEM EXPERIMENTAL RESULTS	49

5.3	PASSIVE IMAGING SYSTEM EXPERIMENTAL RESULTS	54
5.3.1	Target	56
5.3.2	Averaging	59
5.3.3	Edge detection	60
5.3.4	Thresholding	61
5.4	THz ACTIVE IMAGING SYSTEM SIMULATION RESULTS	62
5.4.1	Noise-free simulations	62
5.4.2	Simulations with random noise	65
5.4.3	Effects of the receiver integration time	66
6	CONCLUSION	71
	REFERENCES	73
APPENDICES		
A	RANDOM PATTERN GENERATING CODE	77
B	IMAGE DISCRETIZATION CODE	81
C	CS IMAGE RECONSTRUCTION CODE	83
D	LINE SHIFTING CODE	97
E	PASSIVE IMAGING SYSTEM IMAGE RECONSTRUCTION CODE	101
F	IMAGE AVERAGING CODE	107
G	ACTIVE IMAGING SIMULATION ACCORDING TO PSF	109
H	ACTIVE IMAGING SYSTEM SIMULATION WITH WRONG RECEIVER INTEGRATION TIME	113

LIST OF TABLES

TABLES

Table 5.1 Available results	44
---------------------------------------	----

LIST OF FIGURES

FIGURES

Figure 1.1 Power performance of solid state sources, adapted from [1]	2
Figure 1.2 CS setup for imaging with multiple receiver positions	7
Figure 1.3 CS setup for imaging with single receiver position	8
Figure 2.1 Off-axis parabolic mirror is the part of the mirror lying between P1 and P3	12
Figure 2.2 The Galvo mirrors in our system	13
Figure 2.3 Transmission of various clothing materials at THz range, adapted from [2]	17
Figure 4.1 A CGRS using a single vibrating mirror[3]	32
Figure 4.2 The 340 GHz active imaging system	33
Figure 4.3 The 340 GHz active imaging system from bottom	34
Figure 4.4 The 340 GHz active imaging system from side	34
Figure 4.5 The 340 GHz active imaging system's PSFs	35
Figure 4.6 The 340 GHz active imaging system's central MTF	36
Figure 4.7 The CS imaging system	37
Figure 4.8 One of the random patterns used	38

Figure 4.9 The same random pattern shown in 4.9 is shown here as it was designed	38
Figure 4.10 The target as seen from the receiver's point of view	39
Figure 4.11 Optics of the passive imaging system	41
Figure 5.1 Shifted ideal image	45
Figure 5.2 Evolution of SSIM with increasing number of measurements, using l_1 reconstruction algorithm. (-) ideal data, (o) ideal data + 5 percent noise, (+) ideal data + 10 percent noise, (*) pure noise . . .	47
Figure 5.3 Evolution of SSIM with increasing number of measurements, using TV reconstruction algorithm. (-) ideal data, (o) ideal data + 5 percent noise, (+) ideal data + 10 percent noise, (*) pure noise . . .	48
Figure 5.4 The reconstructed images using 38 measurements and 98 measurements	50
Figure 5.5 Final image obtained using experimental data using l_1 algorithm	51
Figure 5.6 Final image obtained using experimental data using TV algorithm	51
Figure 5.7 Evolution of SSIM with increasing number of measurements, using l_1 reconstruction algorithm. (-) ideal data, (o) experimental data, (*) pure noise	52
Figure 5.8 Evolution of SSIM with increasing number of measurements, using TV reconstruction algorithm. (-) ideal data, (o) experimental data, (*) pure noise	53
Figure 5.9 Reconstructed image	55
Figure 5.10 The target mannequin	56
Figure 5.11 The threat object	57

Figure 5.12 Image obtained from built-in algorithm of the passive imaging system, resolution is 271 x 124 pixels	57
Figure 5.13 Raw image obtained from the passive imaging system	58
Figure 5.14 Averaged image	59
Figure 5.15 Image after NLM algorithm	60
Figure 5.16 Image after thresholding	61
Figure 5.17 First target	63
Figure 5.18 Image obtained from first target	63
Figure 5.19 Second target	64
Figure 5.20 Image obtained from second target	64
Figure 5.21 Third target	64
Figure 5.22 Image obtained from third target	65
Figure 5.23 Porcelain knife	65
Figure 5.24 Image obtained from porcelain knife	66
Figure 5.25 Image obtained from porcelain knife with 10 percent noise	66
Figure 5.26 Image obtained from porcelain knife with 20 percent noise	67
Figure 5.27 Image obtained from porcelain knife with 40 percent noise	67
Figure 5.28 Image obtained with integration time = 30 x ideal	68
Figure 5.29 Image obtained with integration time = 3 x ideal	68

LIST OF ABBREVIATIONS

THz	Terahertz
CS	Compressive sensing
IR	Infrared
MMW	Millimeter wave
TDS	Time-domain spectroscopy
2D	Two dimensional
3D	Three dimensional
FMCS	Frequency modulated continuous wave
PDRO	Phase-locked dielectric resonator oscillator
AMC	Amplifier-multiplier chain
PSF	Point spread function
OTF	Optical transfer function
MTF	Modulation transfer function
SNR	Signal to noise ratio
TV	Total variation
FFT	Fast Fourier transform
CGRS	Confocal Gregorian reflector system
SSIM	Structural Similarity
NLM	Nonlocal means

CHAPTER 1

INTRODUCTION

One of the least investigated regions of the electromagnetic spectrum is the Terahertz range [4]. Although Terahertz (THz) literally means 10^{12} Hz, Terahertz range is used to describe all electromagnetic waves which has frequency between $3 \cdot 10^{11}$ Hz and $3 \cdot 10^{12}$ Hz. This frequency range corresponds to a wavelength range between 1 mm and 0.1 mm. The Terahertz region is bounded by millimeter waves in the lower frequency (longer wavelength) and infrared (IR) in the higher frequency (shorter wavelength). Millimeter waves (MMW) can be generated electronically [5] and infrared waves can be generated optically [6], hence they are well explored and are commonly used in daily life [7, 8].

Being one of the least investigated regions of the electromagnetic spectrum, THz range is also named the “THz gap” [1]. The reason for this “gap” is the fact that THz waves have a too high frequency to be generated electronically and too low frequency to be generated optically. Figure 1.1 shows power performances of various solid-state sources.

Although THz waves are not easy to generate, once generated they are very useful for many applications. One of the applications is characterizing the chemical properties of a target. This application uses the fact that some materials have characteristic absorption lines in the THz range. In order to manipulate this phenomena Terahertz time-domain spectroscopy (TDS) systems are used. TDS uses pulsed signals to characterize the target [9].

Another application of THz waves is THz communication. Due to their higher

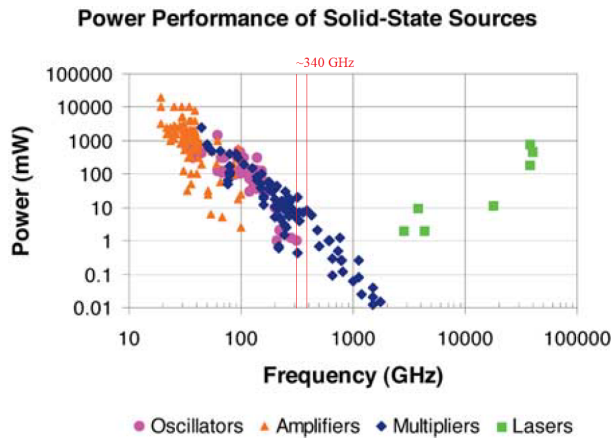


Figure 1.1: Power performance of solid state sources, adapted from [1]

frequencies compared to MMW THz waves are able to transfer data faster. For THz communication systems data transfer rates up to 55 Gigabits per second have been reported [10]. Transfer rates of this order are impossible to be reached by conventional wireless systems since they use microwave frequency band.

The third application of THz waves to be mentioned in this work is the main focus of it: THz imaging. Imaging is the formation of an image. Although homo-sapiens have 5 senses, in long range detection the most efficient of them is seeing. This is because of the fact that sounds, smells, vibrations or tastes of objects cannot travel long distances due to nature of their propagation. This simple range advantage is the reason why one of the most complicated functions of human brain is image processing [11].

1.1 TERAHERTZ IMAGING

THz radiation is not harmful for living organisms [12]. This is because the energy of THz radiation is low compared to harmful radiations, for example x-rays. Although threat detection systems can be implemented using various frequencies, they cannot be used in the field if they are harmful. This is the main advantage of THz imaging. Another advantage of THz imaging is its ability to detect plastic explosives [13]. Conventional metal detectors are able to detect metal threats but they are unable to detect plastic explosives [14].

THz imaging can be separated into two groups as active THz imaging and passive THz imaging. In passive imaging, the source of the received signal is the target itself. Since the THz radiation has a low frequency, and hence a low energy per photon, any object at room temperature emits THz radiation [15]. In active imaging the target is illuminated by THz radiation. For best image quality the receiver in active imaging should only record the incident radiation's reflections, not the radiation by the target. Hence the receivers for these two imaging methods are not necessarily similar.

Imaging a scene in physical world digitally is called digital imaging. Digital imaging has two main steps: encoding and decoding [16]. In the encoding step the data about the scene is collected, that is why encoding is also called the acquisition or image acquisition step. In the second part the encoded data is decoded to create the image. In the decoding step the acquired data is used to reconstruct the image, that is why it is also called the reconstruction or image reconstruction step.

The reconstructed digital images cannot be perfect copies of the physical world. The limitations come from the optical limitations of the imaging system, the electronic limitations of the encoding step and the computational limitations of the decoding step. That is the reason why image reconstruction is followed by, or simultaneously done with, image processing. Image processing is a form of signal processing. Digital signal processing is the processing of digitized discrete-time sampled signals. Possible processes include filters. Although optics of imaging systems and the processes of encoding-decoding have been studied in detail for many years for visible imaging, even visible cameras still use image processing for reconstruction [17]. Image processing is such an integral part of digital imaging the theories related to image processing predates digital imaging systems [18]. THz imaging systems also use acquisition, reconstruction and image processing techniques to create an image.

1.2 IMAGE RECONSTRUCTION

Converting received signals to an image is conventionally the reversing of the process of receiving signals from the scene. Therefore the reconstruction is closely related to the acquisition. The acquisition is done through altering the optical system so to consider different reconstruction techniques one must consider different imaging systems.

One of the most commonly used techniques for imaging is using multiple transceivers [19]. The multiple transceivers are placed on a rectangular array and this array is called an interferometric array since the reconstruction technique is called radio interferometry [20]. Since this method uses interferometry it requires both the amplitude of the received signal and its phase. Radio interferometry combines the received signals' amplitudes and phases to reconstruct the target. This method can be used to reconstruct the target as a two dimensional (2D) image or a three dimensional (3D) surface. This method requires multiple transceivers which makes the cost of building such a system unfeasible. Therefore systems using radio interferometry principles generally have a motion (rotational or linear) to decrease the number of transceivers required. Although it reduces the cost, the motion of the transceivers implies image acquisition time will be greater.

Another image reconstruction technique uses only one transceiver. The imaging is done by focusing the THz beam on the target and scanning the target using scanner mirrors. Scanner mirrors [21] have been reported to acquire images at 2 frames per second. Scanner mirrors are light mirrors which are able to vibrate at frequencies up to 200 Hz. Although 200 Hz is faster than the video rate, these mirrors are unable to image at such high frequencies. The reason for that is, scanner mirrors which operate at high frequencies can only vibrate in one direction. So to acquire a 2D image one needs at least 2 scanner mirrors. If a single scanner mirror is used then the image acquisition rate is reduced to again levels beyond video rate [22]. To the writer's best knowledge, to date no method using scanner mirrors have been reported to acquire images faster than the video rate. To keep the light focused on the target at all scan angles, the optical design should be optimized. Confocal Gregorian dual-reflector system is

reported to optimize image quality on such systems [23]. Image reconstruction in such systems are done combining the mirror position with the signal received. If the received signal's amplitude is the only parameter that is recorded then the reconstructed image is 2D. If the system has a transmitter which emits frequency modulated continuous wave (FMCW) than the imaging can be made 3D, provided that the receiver records both amplitude and phase values of signals [20].

1.3 IMAGE PROCESSING

Digital image processing is used to process digital images. The processing is done by computer algorithms. Digital processing tools used in this work can be deconstructed into three main areas: averaging, edge detection and thresholding.

Averaging is an algorithm which is used to remove noise from an image using convolution of a function with the image [24]. This algorithm is used because reconstructed images may have noises caused by random effects, which cancels out in the average. Therefore averaging each pixel with its neighbourhood removes noise from the image. However this algorithm also blurs the image, making the edges of objects harder to recognize. Therefore edge detection algorithms are also used.

Edge detection algorithms calculates gradients of pixels to determine whether they are on the edge or not [25]. If they are on the edge then the pixels may be recorded if the reason of imaging is to localize a brightness difference in an image. For example in security applications of THz passive imaging possible threats are recognized using the emission differences in the image between possible threats and skin of the person carrying the threat object [26].

Although edge detection algorithms are effective, they can find unrealistic edges for an object if the image has still noises if significant order after averaging. This may be because of noises caused by the system itself, which are not random and hence are not removed by averaging. In these cases an additional step, thresholding, is required for accurately detecting edges. Even if thresholding

will be done after edge detection, still edge detection may be useful. The pixels which are decided to be on an edge may get their gradient with their adjacent pixels enhanced to increase sharpness of the image.

Thresholding is an algorithm used for segmentation of an image. This algorithm is useful for background subtraction and hence edge detection [27]. Although images obtained after these three algorithms are free of many sources of noise, the resulting image may still be useless in real-life applications if the resolution is too low. In such cases images of different resolution are combined using image fusion algorithms.

Image fusion of images with different resolutions require wavelet transforms to be combined with highest efficiency. Wavelet transforms are transformations which represent images in wavelet orthonormal bases, hence representing an image with multiresolution [28]. This is of crucial importance since combining two images of different resolution is computationally infeasible in any other basis. Image fusion is used in THz imaging to combine THz images of low resolution with visible images of high resolution [29]. This is because THz images have a very low resolution which causes problems in localizing the threat, that is imaged on screen, in real world. To help the users of such imaging systems in localizing the threat, imaging systems also include visible cameras. The images obtained from the visible camera and the THz imaging system are fused and then are shown to the user.

Although various imaging systems and image processing algorithms have been developed to produce a cost efficient and effective THz security system, to date no system has been used globally. One of the main reasons of this is the fact that THz imaging systems require long time intervals to form images [30].

Nyquist–Shannon sampling theorem puts a lower limit for the number of measurements required to form an image of given resolution. Recent developments in information theory shows that under some reasonable assumptions even under-sampled data can be used to reconstruct a unique image. This imaging principle is called compressive sensing (CS) [16].

1.4 COMPRESSIVE SENSING

Compressive sensing (or compressed sensing) assumes that if the image one is trying to reconstruct is sparse in some basis, then less measurements than stated by the Nyquist–Shannon sampling theorem suffices. In other words to reconstruct an image of N^2 pixels, less than N^2 measurements is sufficient [16]. The reason for that is the sparsity of the image. A sparse image means an image where some pixels are zero. Since some pixels are not of concern no measurements are required for those pixels [30].

The assumption on the sparsity of the image is a realistic one. Most of the time, one tries to image an explosive or handgun, which corresponds to a small fraction of the total image. Another assumption in some CS algorithms is related to total variation of the image [31]. In these algorithms the image is assumed to be sparse and its variation is assumed to be minimal. Minimization of variation means the sum of gradients of all pixels is minimal.

Although CS assumes some pixels are zero, it does not assume where they are. Therefore a regular pixel-by-pixel imaging is not applicable for CS. There are two different imaging systems used in CS. The first one requires receiver to be placed at more than one position. For the second only one receiver position suffices.

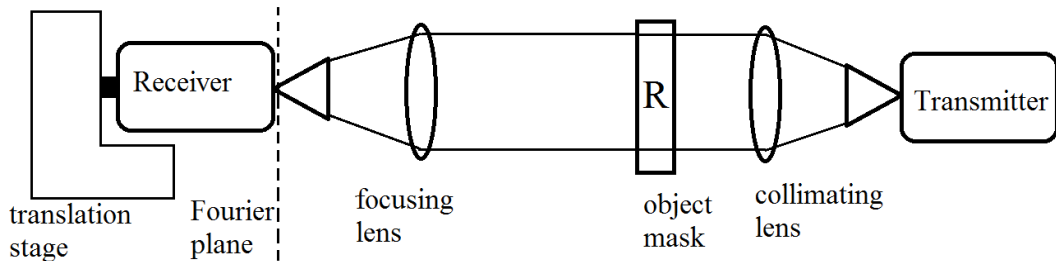


Figure 1.2: CS setup for imaging with multiple receiver positions

In the figure 1.2 an imaging system of first type at transmission mode is shown [32]. Here the waves are generated by a source. Then a converging lens collimates the waves. The object mask transmits at some points and reflects or absorbs at some other. Therefore after the object mask only some of the created waves

can propagate, depending on the object. After the object mask a converging lens focuses the collimated beam on Fourier plane. A receiver measures the amplitude of the signal received at points on the Fourier plane. In order to do that, the receiver is mounted on a translation stage.

In a regular imaging algorithm, the receiver takes measurements at a different position for every pixel that is to be shown in the reconstructed image. When imaging is done using CS less measurements are sufficient to reach the same resolution at the same number of pixels, because the reconstructed image of a sparse object mask is also sparse. Therefore taking any subset of measurements in a regular imaging system suffices for CS [32].

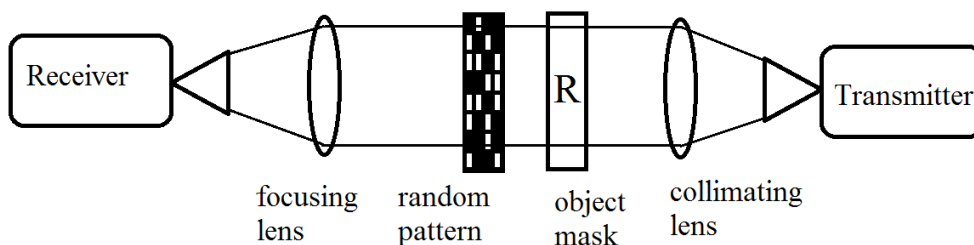


Figure 1.3: CS setup for imaging with single receiver position

In the figure 1.3 an imaging system of second type at transmission mode is shown [33]. Here the waves are generated by a source. Then a converging lens collimates the waves. The object mask transmits at some points and reflects or absorbs at some other. Therefore after the object mask only some of the created waves can propagate, depending on the object. The object mask is the image to be reconstructed. After the object mask there is another mask, called random pattern. This mask also blocks some of the waves. This random pattern is known. Therefore the aim of the imaging system is to only reconstruct the object mask, not the random pattern. After the random pattern a converging lens focuses the collimated beam on the receiver. The receiver is fixed.

In a regular imaging algorithm to reconstruct the image (object mask) the number of random patterns to be placed should be at least the number of pixels in the image. In the trivial case the random patterns should have only one opening

where the waves can pass, this opening should have the size of a pixel on the image and all matrices have the opening at different positions. When imaging is done using CS, measurements using less number of matrices suffice, since if the object mask is sparse, then the reconstructed image is also sparse [33].

The main goal of this thesis is to understand the effects of various parameters that are used in the formation of images in an active THz imager based on frequencies near 340GHz. Those parameters include but are not limited to optical system design, target geometry, receiver performance, image reconstruction and image processing tools. The designed optical system was simulated and the effects of scanning the target field were assessed in the simulated images of various targets. The results show that reliable target detection will be aided by post-processing the images using various techniques.

In the second chapter, background related to THz imaging is given. Similarly, third chapter includes background related to simulations of active imaging systems. Chapter four describes all the imaging systems that are used in this thesis. Chapter five includes all the results from the experiments and the simulations. Finally, chapter six concludes this thesis with a summary.

CHAPTER 2

THEORETICAL BACKGROUND OF THZ IMAGING

Active THz imaging systems have four main components: source, optical instruments, target and receiver. For passive imaging systems there is no source since the target acts as a source. In the first four sections of this chapter those components are explained in detail. The last three sections explain working principles and sources of errors in THz imaging systems.

2.1 SOURCE

The source that will be used in 340 GHz imaging system includes a Phase Locked Dielectric Resonator Oscillator (PRDO), an Amplifier / Multiplier Chain (AMC) and a WR2.8DH Horn. For imaging performance not all the electronic details of these components are relevant. The most important factors for this work are stability of the output frequency and power and optical properties of the horn.

Since 340 GHz is a very high frequency to be directly generated electronically, the source uses multipliers to reach 340 GHz. The first component is a PDRO. PDRO uses a crystal reference to minimize fluctuations in the output frequency. The stability is given by the producer as $\ll 1$ parts per million per Kelvin (ppm/C). That means when the temperature changes by 1 degree Kelvin, the output frequency will change less than 1 in a million. The output frequency of the PDRO used in system has an output frequency of 14.17 GHz. The AMC multiplies and amplifies the signal in a specific order. All multipliers are Schottky diodes. First the signal is multiplied by 3, then it is amplified, then it is

multiplied by 2 for three times. This can be formulated as $x^3/\text{Amp}/x^2/x^2/x^2$. After these operations the signal reaches 340 GHz. Since Schottky diodes decrease the output power, the output power after AMC is also of interest. The producer claims the output power to be greater than 10 mW. The results of our measurements with Goly cell found the output power to be near 7mW. Goly cell is explained in detail in the receivers section.

2.2 OPTICAL INSTRUMENTS

The optical instruments used in the described system includes off-axis parabolic mirrors, plane Galvanometer mirrors (Galvo mirrors), off-axis elliptic mirrors and directional couplers.

Off-axis parabolic mirrors are one of the mirrors which are most commonly used in daily life as satellite dishes.

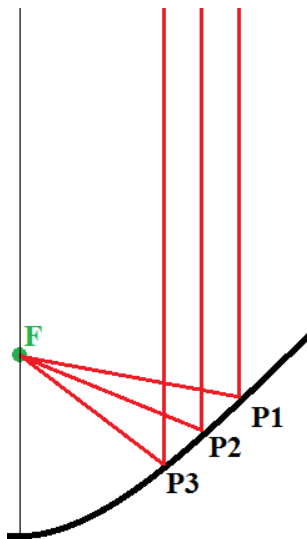


Figure 2.1: Off-axis parabolic mirror is the part of the mirror lying between P1 and P3

Off-axis parabolic mirrors are obtained from parabolic mirrors by removing some parts of the parabolic mirror, including the mirror's vertex. For example satellite dishes have a receiver which would cast a shadow on the vertex of the mirror. Therefore the vertex point of the mirror would not receive any signal from the

satellite, hence this point would be a waste in production. To minimize the cost, this point is removed from the dish. Similarly, in optical systems it is customary to use off-axis parabolic mirrors in order to minimize the cost of the setup.

Off-axis elliptical mirrors are obtained from elliptical mirrors by removing some parts of the elliptical mirror, including the mirror's vertex. They are used for very similar motivations.

Galvo mirrors are galvanometer based vibrating mirrors which are driven by motors. Their vibrating frequency can reach up to 1 kHz but of course the highest possible frequency depends on the inertia of the mirror. In the system that is being explained, the Galvo mirrors should be of at least 50 mm in diameter. Therefore the highest possible frequency for the system is less than 1 kHz. The Galvo mirrors are able to vibrate only in a single direction. Therefore to scan a two dimensional (2D) image at least two Galvo mirrors are required. These optical instruments can be used to image a scene in conjunction as shown below.

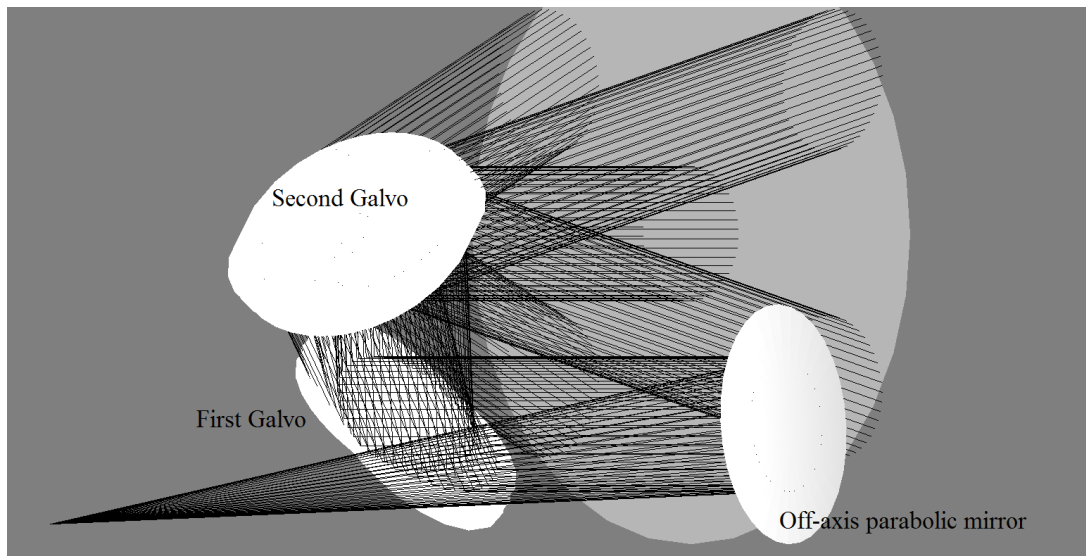


Figure 2.2: The Galvo mirrors in our system

Directional coupler is a waveguide coupler. In our system the waveguide coupler is used to couple the wave-guides between transmitter, receiver and the horn. The directional coupler is directional, so that, when the signal comes from the

source, most of the signal is transmitted to the horn. When the signal is received at the horn, most of it is transmitted to receiver. Naturally there is some leakage in the directional coupler. The noise level from this leakage is eliminated using pulsed signals from the transmitter and considering the receive times of the receiver in image reconstruction.

2.2.1 Point Spread Function

Point Spread Function (PSF) measures the response of an imaging system to a point source. It is the spatial domain version of the Optical Transfer Function (OTF). In other words, 2D Fourier transform of PSF yields OTF. Modulation Transfer Function (MTF) is the real component of (OTF). The complex parts of PSF and OTF only affect the phase of the signal. As we will discuss in the receiver section, we are not interested in the phase. Therefore from this point on, PSF will denote only the real part of the point spread function.

$$MTF = \mathcal{F}(PSF)$$

PSF is of great importance to understand performance of an optical system because it determines the resolution of the imaging system. Even if an optical system is ideal, still the PSF of it would not be a single point because of the limitations from diffraction.

Convolution is a mathematical operation which applies on two functions to generate a third function. PSF can also be defined using convolution. If a 2D target is defined by $T(z, w)$ and a PSF is defined by $PSF(u, v)$ then 2D convolution of these two gives the image, $I(x, y)$, that will be obtained from this imaging system:

$$I(x, y) = (T(z, w) * PSF(u, v)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} T(\alpha, \beta) PSF(\alpha - x, \beta - y) d\alpha d\beta$$

Here, magnification is assumed to be 1. If magnification is not 1, but M, then

the formula is modified as:

$$I(x, y) = (T(z, w) * PSF(u, v)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} T(\alpha, \beta) PSF(\alpha - x/M, \beta - y/M) d\alpha d\beta$$

If one replaces PSF with delta functions than it is trivial to confirm the magnification. Larger PSF causes blurry images. Therefore it is natural that resolution and PSF are related.

It is appropriate here to discuss that the assumed target is noise free throughout this work. All the noises are added after the PSF related operations are completed. Therefore the T given above is completely noise free.

2.2.2 Spatial Resolution

Spatial resolution of an image is the number of independent pixels the image contains. The independence here can be defined using PSF. If two pixels of an image corresponds to distances smaller than the PSF of the system, then those pixels are not independent of each other. That is because they share some of the information they display in common. Although this definition of spatial resolution is well-defined in theory, practically it is meaningless. Most PSFs are Gaussian functions, therefore they never become zero. Therefore generally a cut-off is placed to decide the pixel size.

It is discussed that pixel based images can be enlarged to achieve higher resolution [34]. Algorithms providing such processes are called super-resolution algorithms. These algorithms use guesses for missing frequency components of the enlarged images. Therefore they are post-processing applications. Another post-processing application which also yields super-resolution images is based on moving the imager [35].

As mentioned above, MTF is the Fourier transform of PSF. Combining PSFs of many optical instruments requires more computation than combining MTFs. Therefore in optical design one generally focuses on MTF. In the system that we discuss in this work the sizes of the pixels are decided using the 50 percent MTF as a limit. As shown by super-resolution algorithms, to define a definite pixel

size is meaningless. It is a must, however, since imaging algorithms require a defined pixel size. Having too many pixels (small pixel size) in an image implies slower operation but better resolution. Having too few pixels (large pixel size) implies faster operation but worse resolution. Therefore a trade-off should be reached. It is a customary limit to choose percent 50 MTF as a limit in imaging applications.

MTF values of an imaging system are bounded by diffraction limit. Therefore longer wavelengths have worse spatial resolution in general.

2.3 TARGET

The intended target of the imaging system described is human body. As discussed in the introduction the advantage of THz imaging is that it is not harmful [12]. However since the system will be used on dressed people it is of great importance to understand the transmission coefficients of various clothing materials at 340 GHz.

Figure 2.3 shows transmission of various clothing materials at 340 GHz [2]. In the figure, 0 dB corresponds to no barrier. The thickness of the samples used in measurement was not fixed. The thickness was chosen considering the common clothing materials. For example the leather was 0.75 mm thick and so on... Please see original work for further technical details. As can be seen from 2.3 with increasing frequency transmission becomes smaller. Therefore longer wavelengths are better for penetrating through clothing. As mentioned in the previous section, shorter wavelengths yield better resolution. Therefore the operating frequency of the system described in this thesis is determined with a trade-off: shorter wavelengths are better for resolution while longer wavelengths are better for penetration.

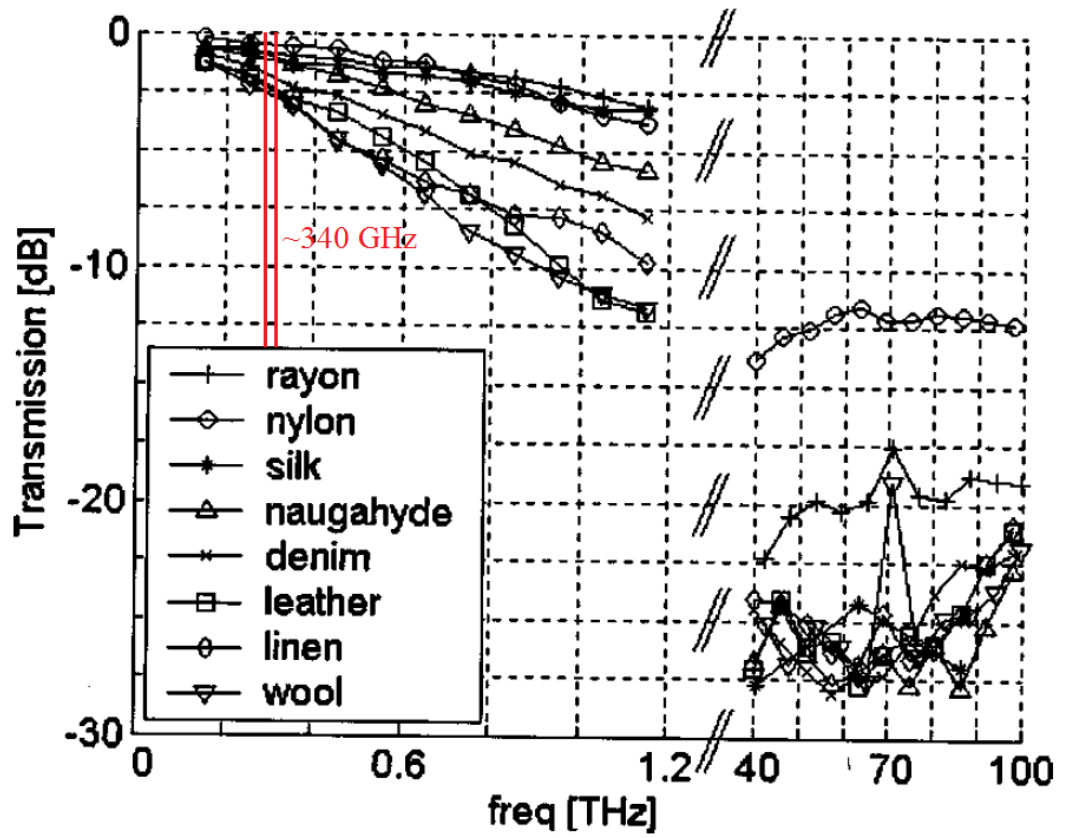


Figure 2.3: Transmission of various clothing materials at THz range, adapted from [2]

2.4 RECEIVER

The receiver used in the 340 GHz system consists of a Mixer Amplifier Chain (MixAMC) which has its own local oscillator (LO) and shares the WR2.8DH Horn with the source. The reason of sharing the horn is for optimum signal level. The optical design requires the source and the receiver to be coincident for receiving maximum signal. The MixAMC mixes received signal and multiplied LO signal to down convert the received signal. The LO signal in the receiver is multiplied similar to the transmitter setup except for the fact that the mixing is done at the subharmonic. First the signal from LO is multiplied by 2, then it is again multiplied by 2 then it is amplified, then it is multiplied by 3. This can be formulated as $x_2/x_2/Amp/x_3$. The down converted signal is called Intermediate Frequency (IF) signal. The reason for using IF signal and not the received signal to form the image is because of the imaging frequency. 340 GHz is a very high frequency for any electronic component to be able to manipulate, therefore the received signal has to be down converted.

Another receiver is used multiple times throughout the thesis. This other receiver is called a Golay cell. Golay cell consists of a gas container with two ends; the first end receives the signals and the other end is closed with an elastic membrane. The received signals heats the gas and thus causes vibrations on the membrane. The vibrations on the membrane are measured, thus enabling the user to learn the power input on the Golay cell.

2.5 NOISE

There are several possible sources of noise for THz imaging. One of the most important source of noise is multiple reflections from the target. Although multiple reflections effect the performance of active imaging systems severely, they are unavoidable since the source of reflection, target, is unknown in any imaging application.

Another source of noise is the black-body radiation. Since THz waves are of very

low energy, all objects in room temperature emit THz waves. Therefore in an active imaging system some of the received signal is not the reflected signal but is the black-body radiation. The amount of radiation at a given temperature at a given frequency can be found using Planck's law [36]. The Planck's law states that

$$B_\nu = \frac{2h\nu^3}{c^2} \frac{1}{e^{\frac{h\nu}{k_B T}} - 1}$$

where B_ν is spectral radiance, T is absolute temperature, k_B is the Boltzmann constant, h is the Planck constant, c is the speed of light in vacuum and ν is the frequency. To calculate the total energy radiated in a given frequency range one needs to integrate B_ν . Calculation shows that the total radiation between 339 GHz and 341 GHz is equal to $0.2596\mu W/m^2$. This number should be multiplied with the surface area of the receiver's horn, therefore the noise is in the order of nanowatts.

The noise level of receiver is determined by the lowest signal that the receiver can distinguish from background. The imaging performance of an imaging is closely related to its signal to noise ratio (SNR). The signal to noise ratio is the ratio of received signal and the noise level. A low SNR means a low contrast level in the obtained image.

2.6 ENCODING

The encoding and decoding processes are final processes of imaging therefore they require all components to be ready to perform. Unfortunately this was not the case for the system that is being described. Therefore encoding and decoding are investigated not for the 340 GHz active imaging system but another imaging system, called Compressive Sensing (CS) imaging system.

2.6.1 CS Imaging System

The source that was used in the CS imaging system was very similar to 340 GHz source, the only difference was in frequency. The output frequency of the source

used in CS imaging system was 120 GHz. The optical instruments were placed as described in 1.3. The lenses were Teflon lenses. The receiver used was Golay cell receiver. The Golay cell receiver was described in the receivers section of this chapter. The target and the object masks were obtained using laser cutting aluminium sheets.

2.6.2 CS Encoding

In CS encoding and decoding processes are not one to one. Therefore the reconstruction process relies on optimization algorithms [30]. An image is a two dimensional array of numbers, where the numbers correspond to intensity of signals at the pixels. Assume the image is square, that is number of pixels on rows of the image are equal to those on columns. Then the image is a matrix of size $N \times N$. Any matrix of size $N \times N$ can be expressed as a vector of size $N^2 \times 1$. An example of doing this is placing the second column of x under the first one and so on. Therefore any 2D image can be expressed as:

$$x = \sum_{i=1}^M \theta_i \cdot \Psi_i$$

Here x is a two dimensional square image which is expressed as a vector of size $N^2 \times 1$, M is the number of measurements, θ_i are coefficients of Ψ_i and Ψ_i are column vectors of size $N^2 \times 1$. In other words Ψ_i are basis vectors and θ_i are coefficients, so the above formula is the representation of x in some basis Ψ_i . In the case of the imaging setup shown on 1.3, Ψ_i are the random patterns which are placed in the setup. So CS uses the random patterns as bases to represent the image, x . In the encoding process the receiver records the amplitude of the signal received. The measurements can be expressed as:

$$y = A \cdot x$$

Here y lists measurements, A is the measurement matrix and x is the image. y has dimension $M \times 1$ since there are M measurements done. x was given to be a matrix of size $N^2 \times 1$ therefore the measurement matrix has size $M \times N^2$.

In the case of the imaging setup shown on 1.3, the measurement matrix is the matrix obtained by placing transpose of the second random pattern under the first one and so on. Combining the equations for y and x we get

$$y = A \cdot \left(\sum_{i=1}^M \theta_i \cdot \Psi_i \right)$$

In the case of the imaging setup shown on 1.3, i^{th} row of A is Ψ_i^T where T denotes transpose. Therefore the matrix multiplication can be simplified for y_j , the j^{th} row of y as:

$$y_j = A_{j^{th}row} \cdot \left(\sum_{i=1}^M \theta_i \cdot \Psi_i \right)$$

$$y_j = \Psi_j^T \cdot \left(\sum_{i=1}^M \theta_i \cdot \Psi_i \right)$$

Since θ_i are only numbers we can express the same formula as:

$$y_j = \left(\sum_{i=1}^M \theta_i \cdot \Psi_j^T \cdot \Psi_i \right)$$

It is important to note that if the basis vectors are orthogonal to each other, then the $\Psi_j^T \cdot \Psi_i$ term becomes Kronecker delta and reconstructing the image would be trivial. However this is not the case for CS. In CS the bases are not orthogonal to each other on purpose. After all the measurements are done, the computer has the measurements, y_i ; measurement matrix, A ; and hence base vectors, Ψ_i . Using these information to find x is called reconstruction.

2.7 DECODING

Decoding the encoded data is called reconstruction in imaging applications. In CS the decoding process is an optimization process, due to assumptions of CS. In this work two different reconstruction algorithms are used. The algorithms are called l_1 minimization and total variation (TV) minimization.

The encoding assumes sparsity in target. Therefore it is normal to expect that

the reconstruction algorithm should also consider sparsity. This is not the case in both of the algorithms, because sparsity implies value of components of the image matrix being zero. Choosing which components should be zero is a very time consuming problem for computer. Therefore for imaging systems with realistic image reconstruction timing, sparsity is not a condition in reconstruction. Instead they use l_1 minimization.

l_1 is a metric. For any real number x and y the distance between those two is defined as the absolute value of their difference ($|x-y|$) in l_1 metric. Minimizing an image in l_1 implies forcing it to have minimum value possible in l_1 metric. Without constraints the answer is trivially zero, therefore constraints are a must for reconstruction. The constraints are obtained using the measurements, y_i .

l_1 minimization method given as,

$$x = \arg \min \|x\|_1 \text{ s.t. } y = Ax$$

Finds the solution, x (image matrix) that is a solution for $y = Ax$, where y is the measurement and A is encoding matrix, subject to the condition that,

$$\|x\|_1 = \sum_i |x_i|$$

is minimum[37].

Here x is the image matrix in vector form, y is the measurement, A is the measurement matrix and *argmin* means choose the x with minimum l_1 metric of all possible x 's. There are more than one x satisfying the $y = Ax$ constraint, since A is not a square matrix. The l_1 minimization problem can be solved in polynomial time, therefore is a feasible way to reconstruct image. This algorithm works in reconstructing the image, because of all the possible images, the one with smallest sum of components is the closest one to a sparse image.

There are variations of l_1 minimization algorithm. All those variations assume new constraints on the image. The one we are interested is called TV minimization. In TV minimization, another constraint is imposed on the reconstruction algorithm. This new constraint is about sum of the gradients of the pixels in the

image. If we not only assume that the image is sparse but also there are only a few objects in the image, then the image should be minimal in gradient. That is because the gradient is greatest on the boundaries of objects. Therefore the less objects the less the boundaries and hence the less the sum of the gradients. Mathematically this can be formulated as:

TV, given as,

$$x = \min \text{TV}(x) \text{ s.t. } y = Ax$$

finds the solution, x subject to the constraint.

$$\text{TV}(x) = \sum_{ij} \sqrt{(D_{h;ij}x)^2 + (D_{v;ij}x)^2},$$

s is minimum[38], where

$$D_{h;ij}x = \begin{cases} x_{i+1,j} - x_{ij} & i < n \\ 0 & i = n \end{cases} \quad D_{v;ij}x = \begin{cases} x_{i,j+1} - x_{ij} & j < n \\ 0 & j = n \end{cases}.$$

Here x is the image matrix in vector form, y is the measurement, A is the measurement matrix and $\min \text{TV}(x)$ means choose the x with minimum total variation of all possible x 's. There are more than one x satisfying the $y = Ax$ constraint, since A is not a square matrix. The TV minimization problem can also be solved in polynomial time, therefore is a feasible way to reconstruct image. This works in reconstructing the image, because of all the possible images, the one with smallest sum of gradients is not only of minimal variation but also of minimal pixel values. Hence it is natural to expect TV to outperform l_1 minimization in general. In fact this is the case, therefore most of the CS imaging systems in literature only use TV minimization.

CHAPTER 3

THEORETICAL BACKGROUND OF ACTIVE IMAGING SYSTEM MODELING

In the previous chapter the theoretical background of THz imaging was investigated in reality. Modeling the explained reality in computer is not trivial, since reality has the advantage of using continuous functions while computers can only manipulate discrete data. Therefore all the components explained in the previous chapter should be manipulated analytically for one to be able to simulate them using a computer.

In this chapter, discretization of two main properties are investigated: PSF and noise. The other parameters are trivial to discretize. For example output power of the source is in reality a real number. However as soon as one measures it, it becomes discretized; because of the error in measurement, one can safely assume the output power to be a number with reasonable significant figures. However this is not the case in PSF. There are too many optical components and too many mirror positions to try to measure PSF experimentally. Moreover to produce optimized optical components one must have an understanding of their PSF.

The noise in an imaging system is also not trivial to model, because as discussed in previous chapter there are numerous sources of noise. Moreover, due to nature of noise, noise is not a predictable parameter. Therefore it is not trivial to model it in a computer.

3.1 PSF

As mentioned in previous chapter PSF of an optical system is of great importance in an optical system's performance, especially in resolution. Therefore designing the optical instruments with ideal PSF values is important for an optical system. To design the instruments, ZEMAX, a commercial optical design tool was used. ZEMAX has built in functions for calculating PSF. Without understanding how those functions work it is impossible to understand the limitations and benefits of them.

Before giving more detailed information related to computation of PSF it is appropriate to explain how ZEMAX works in brief. ZEMAX defines optical instruments sequentially: every optical element has an entrance and an exit pupil. The entrance pupil of an optical element is the exit pupil of the previous one.

3.1.1 Modeling PSF of a single optical element

ZEMAX can model PSF of an optical element using three different ways. The first way is using geometry of the incident and reflected light. This is of little interest in this work since the resolution of our system is limited by physical optical phenomena.

The second way uses Fourier transform of rays to calculate PSF. This method is called FFT PSF. In this method, a grid of rays are traced from the entrance pupil to the exit pupil of the optical element. The number of rays in the grid are chosen by the user. During the tracing of the rays, the optical path each ray travels is calculated independently. After tracing, the grid at the exit pupil is analysed for each ray and then a complex amplitude is calculated for each of them. The amplitude is complex in order to include the phase of the rays. After all computations for all the rays are completed, the Fourier transform of the grid is calculated, scaled and squared to obtain PSF.

The Fourier transformation is taken using Fast Fourier Transformation (FFT)

algorithms. Therefore the uniformity of the samples is crucial for a reliable result. However this is not the case in general. For example even if the rays are uniformly distributed on the entrance pupil, they may be localized on the exit pupil, therefore making FFT algorithms unreliable. For example in an off-axis parabolic mirror, the entrance pupil and the exit pupil are tilted because of the tilt of the mirror, making FFT PSF unreliable. Therefore the most reliable PSF calculation for off-axis mirrors is the third way.

The third way of calculating PSF using ZEMAX is using the Huygens PSF option. This option uses the Huygens principle, as the name suggests. Again a number of rays specified by the user are traced from the entrance pupil to the exit pupil. In contrast to the FFT PSF method, this method traces the rays not as rays but as plane waves. The rays at the entrance pupil are converted to plane waves according to Huygens principle, therefore all the rays correspond to one plane wave with a definite amplitude, phase and direction. To calculate the PSF all the plane waves are summed on the exit pupil. Therefore this is the slowest method of all three. However it is the most reliable.

3.1.2 Modeling PSF of an optical system

In the previous subsection PSF calculation for a single optical instrument was investigated. However imaging systems consist of multiple optical instruments in general. Therefore modeling PSF of an optical system requires combining PSFs of multiple optical instruments.

As seen in the previous chapter, combining multiple PSFs are trivial in ZEMAX since one optical instrument's exit pupil is the entrance pupil of the next one. However obtaining the final PSF is not of great significance since the performance of the imaging system is generally measured using MTF, not PSF. Therefore after calculating the final PSF it is customary to take its Fourier transformation to obtain MTF.

In ZEMAX the obtained MTF is called geometric MTF if the PSF was calculated using geometric PSF. If PSF was calculated using FFT method then the resulting

MTF is FFT MTF. The one that was used in this study is in the third method, Huygens MTF which is calculated by propagating wavefronts from the source to the image plane to obtain PSF and the applying Fourier transformation to get Huygens MTF. The pixel size of our imaging system is determined by this MTF.

The wavefront in reality is a continuous wavefront and therefore is impossible to be modeled in a computer. Therefore ZEMAX discretizes the continuous wavefront to small discrete wavefronts and propagates them separately, only to sum in the end to obtain an approximation to reality. This calculated wavefront determines the signal that is sent to the target. If the reflection coefficient of the target is known then this means one also knows the signal to be received. Therefore modeling PSF one is able to model an active imaging system.

In the optical system shown in figure 2.2 every different configuration has a different PSF, therefore to model an imaging system with vibrating mirrors one needs more than one PSF. In the imaging system described in this paper, one needs at least 900 different PSF values to model the image. However calculating 900 Huygens PSFs is a time consuming process. Therefore interpolation is used to calculate some of the PSFs. Detailed information will be provided in chapter 5.

3.2 Noise

The major sources of noise in the explained imaging system can be listed as noises from the environment and the noises from the receiver. Although both has a random nature they may affect the image reconstruction in different ways because of the reconstruction algorithm used. Therefore they are investigated separately.

Modeling noise is important to understand the robustness of reconstruction algorithms. An imaging system that only performs well in laboratory conditions is of little interest in practical problems therefore robustness of the imaging system is also an important parameter for this work.

In order to explain adding noise to a model, one needs to describe the model. The model imaging system has a defined target and defined optics. Therefore the signal profile on the target is defined. When target is also defined the received signals can be calculated. However in reality there are inevitable sources of noise which need to be modelled to simulate a realistic received signal. Without any noise added, the calculated signals to be received are called ideal signal, since they are noise free. In the following subsections the nature of the noise to be added and how to add it to signal are discussed.

3.2.1 Modeling noises from environment

The noises from the environment were listed as black-body radiation and multiple reflection in the previous chapter. Since multiple reflections are dependent on the geometry of the target and the target will be unknown in real-life operation of this system, one can model noise from multiple reflections as a random noise. Similarly, temperature fluctuations on the target can cause noise in the image reconstruction. Again the fluctuations are not predictable beforehand, therefore the noises caused by black-body radiation are also of a random nature.

Random noise from the environment is modeled in this work by adding random levels of signals to the received signals during the simulations of image reconstruction. Since temperature fluctuations can be in both positive and negative direction, the added random signal was not forced to be positive. The effects of this random noise is discussed in chapter 5.

3.2.2 Modeling noises from receiver

In the previous chapter it was stated that the noise level effects the quality of the reconstructed image in a direct way. The SNR of the imaging system limits the possible contrast that can be captured in images.

The noise level in the receiver defines the step size for contrast levels. For example if one has a noise level of 0.01 mW and a peak signal of 1 mW then the signal can consist of maximum 100 steps. Since any difference less than 0.01

mW would not be detectable, due to noise.

On the computer screen most of the images have a contrast level of 256. This means a computer screen can show maximum 256 shades of grey. However if SNR is lower than 256, it is impossible to assign 256 levels of contrast to this signal. This implies reducing the contrast level of the reconstructed image.

Another possible problem caused by the noise from receiver is fluctuation beyond the expected level of noise. In this case the shown pixel will display a wrong shade of grey. Since this noise is unpredictable and can both increase and decrease the signal level, it is also a random noise. Therefore it can be modelled as mentioned in the previous subsection.

The final possible problem from noise in the receiver that will be discussed in this work is related to integration time of the receiver. If the signal is changing faster than the receiver can respond, then there can be some systematic noise in the received signal. For example if a system with a slow receiver is scanning a point like source, the corresponding image will not be a point but it will be a line. Since even after scanning the point like source, the receiver will be unable to return back to reading zero. This is the only source of systematic noise that is discussed in this work. This noise can be modelled easily by adding the signal that would be received ideally itself after calculating a shift, hence simulating an experiment with a slow receiver.

In conclusion most of the possible sources of noise for the system discussed is of random nature and are modelled with adding the ideal signal a random noise signal. The only systematic noise that is included in this work is the receiver with long integration time, which is modelled with adding the ideal signal itself after a delay.

CHAPTER 4

IMAGING SYSTEMS

The 340 GHz active imaging system, which is the main focus of this work, is described in the first section of this chapter. As mentioned before, encoding and decoding are investigated using a setup based on compressive sensing principles. This setup is described in the second section. Finally a passive imaging system which was used in investigating post-processing algorithms is described in the third chapter. In summary, this chapter covers all the imaging system details required to interpret the results given in chapter 5.

4.1 340 GHz ACTIVE IMAGING SYSTEM

The main purpose of 340 GHz active imaging system is to complete security checks on high-security points like airports. Therefore the system is required to have a good resolution to be able to detect threats with high efficiency and a large enough field of view to scan torso of a person. The optical design most suitable for this task utilizes a confocal Gregorian geometry. Figure 4.1 shows an example of a confocal Gregorian reflector system (CGRS) [3]. In this figure there is only one mirror which is able to rotate both in x and y directions. However this is not the case in the 340 GHz system. As discussed before, since Galvo mirrors are able to rotate in only one direction, two Galvo mirrors are used in the system. Since visualization of a system using 2 Galvo mirrors is not easy, CGRS is explained using figure 4.1.

In the system shown in figure 4.1, the signal is transmitted from transceiver. The

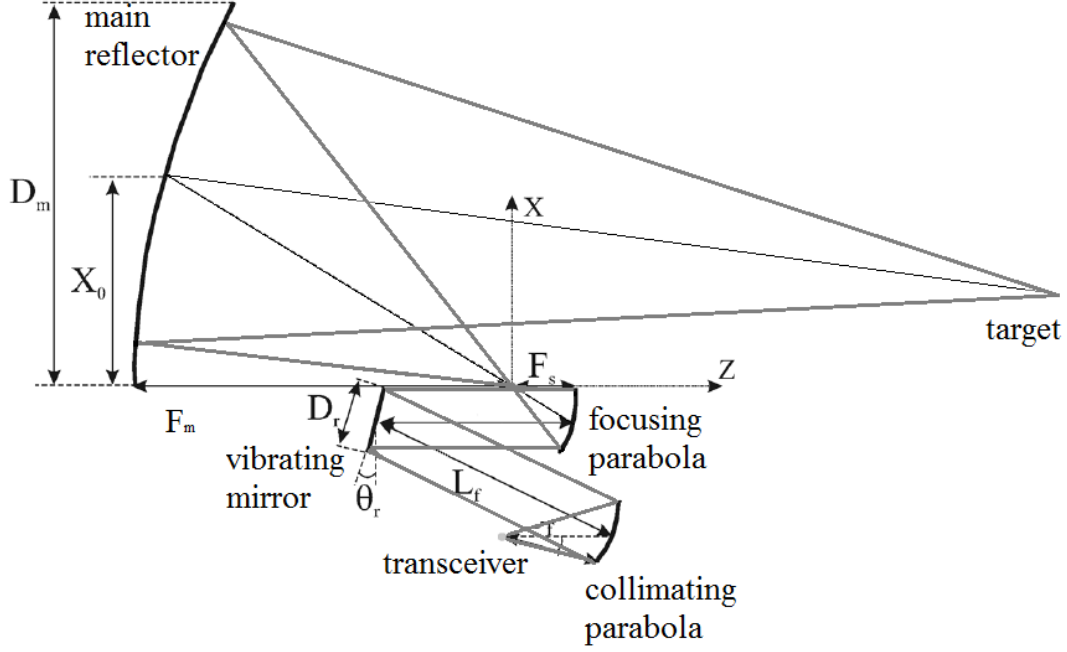


Figure 4.1: A CGRS using a single vibrating mirror[3]

signal has a Gaussian distribution. The first optical element, the collimating off-axis parabolic mirror, collimates the light. The collimated light is steered using a vibrating mirror. Then the focusing off-axis parabolic mirror focuses steered light on a point. The point varies with different configurations of the vibrating mirror. Finally the main reflector, an off-axis elliptic mirror, focuses the signal on target. At every different configuration of the vibrating mirror, the signal is focused on a different point on target. The total field the signal covers on the target is called the field of view.

The CGRS is an optimized system for scan based imaging. The focusing parabola and the elliptical mirror system enables a robust imaging performance. However it has some disadvantages. For example the elliptical mirror's surface has to be precisely shaped for best performance[3]. Moreover if two scanning mirrors are used then the time geometry of the system causes aberrations on the field of view. This aberrations are the main point of interest of the simulation that are described in the chapter 5.

In the 340 GHz active imaging system described in this work, the field of view

is 50 cm x 50 cm. The system was designed and optimized using ZEMAX. The optimization was achieved using the built-in optimization functions of ZEMAX. System was optimized, using merit functions, for maximum resolution throughout the field of view. The main principle of optimization lies in calculation of PSF, which was explained in detail in the previous chapter. The details of the optical system are not given as at the time of this thesis the project was ongoing. Interested readers should contact the author for further information.

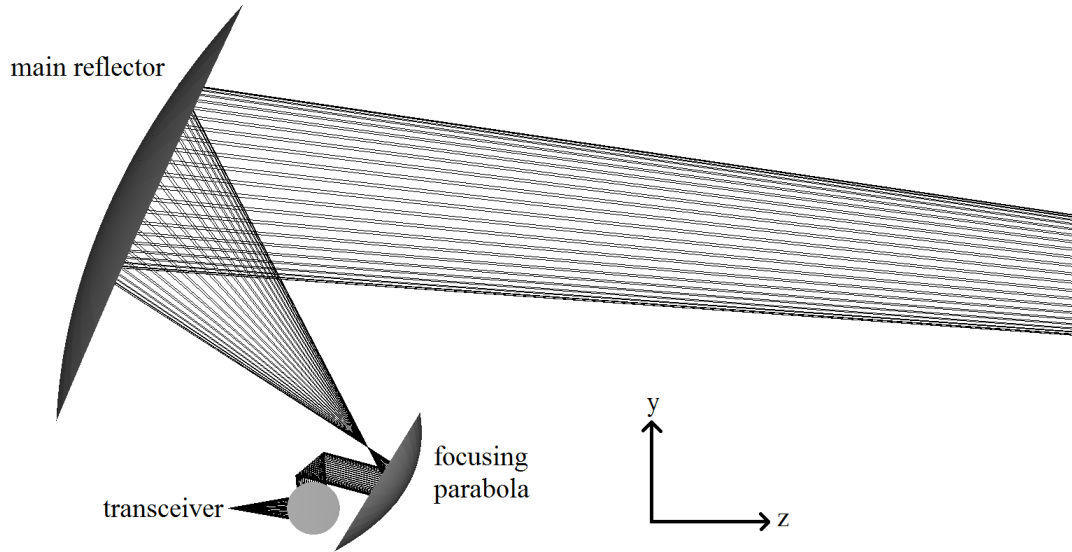


Figure 4.2: The 340 GHz active imaging system

Figure 4.2 shows the optical layout of the 340 GHz active imaging system. Here the main reflector, the focusing parabolic mirror and the transceiver are shown clearly. However the collimating parabolic mirror and the Galvo mirrors are not seen due to the orientation, therefore another view is also given. Figure 4.3 shows the same setup in another angle. In this figure the placement of collimating parabolic mirror is clear, however Galvo mirrors are still not clear. Therefore a third figure is given. In the third figure, 4.4, Galvo mirrors are shown clearly. The propagation of signal for different angles of the Galvo mirrors are also shown in the same figure. As can be seen, the alignment of Galvo mirrors and the focusing parabola makes it impossible to focus the signal perfectly, due to spherical aberration. Effects of this aberration on imaging system performance is investigated in detail in chapter 5.

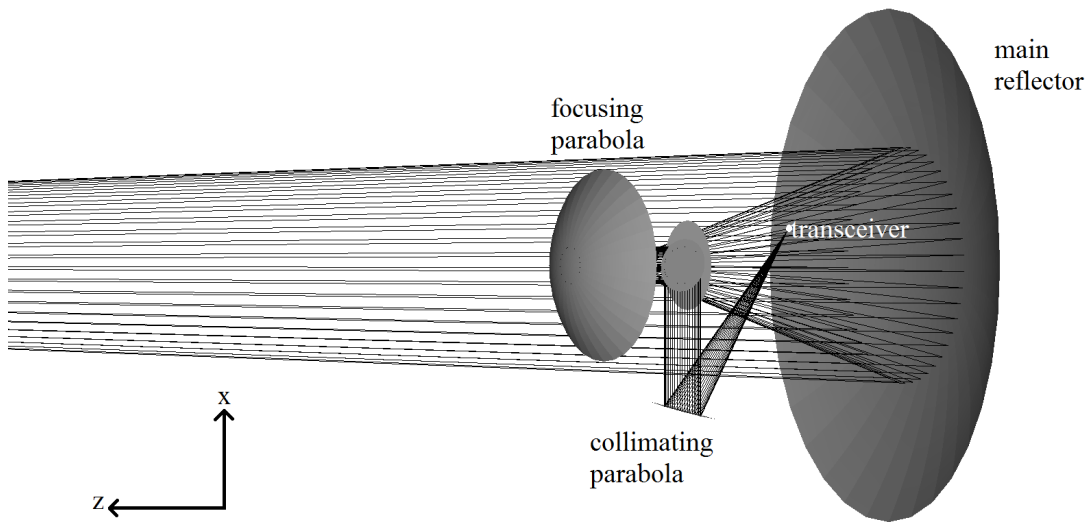


Figure 4.3: The 340 GHz active imaging system from bottom

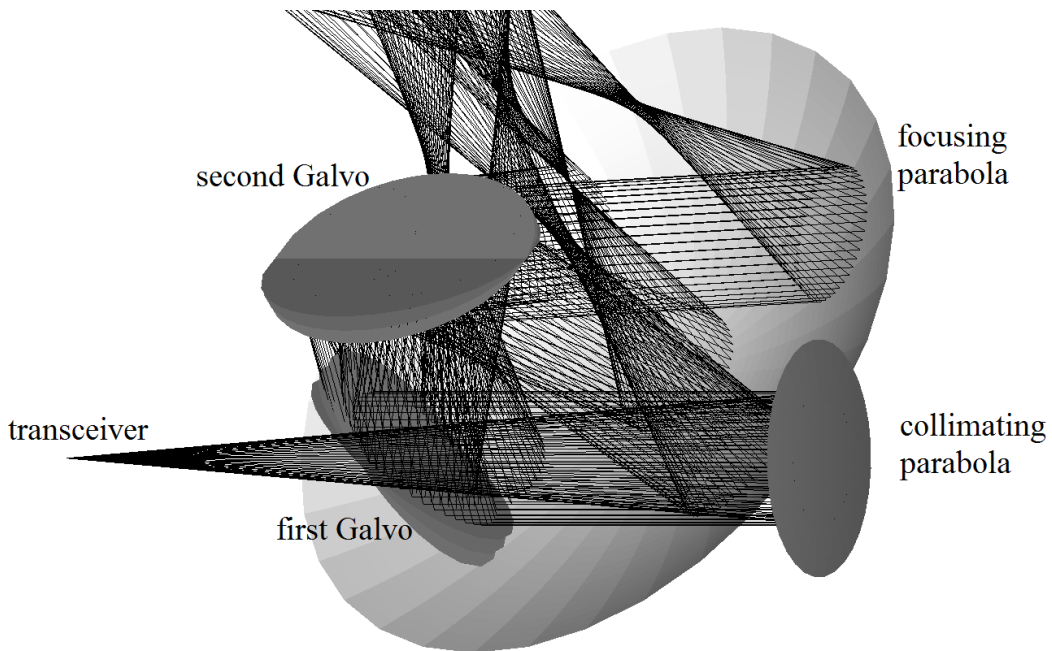


Figure 4.4: The 340 GHz active imaging system from side

The PSFs obtained using Huygens principle from ZEMAX are shown in figure 4.5. In the figure the top left PSF is the PSF at the top left corner of the field of view of the system. The central is the central and so on... The PSFs are all shown for a square of 10 cm x 10 cm. So in total the figure corresponds to 30 cm x 30 cm.

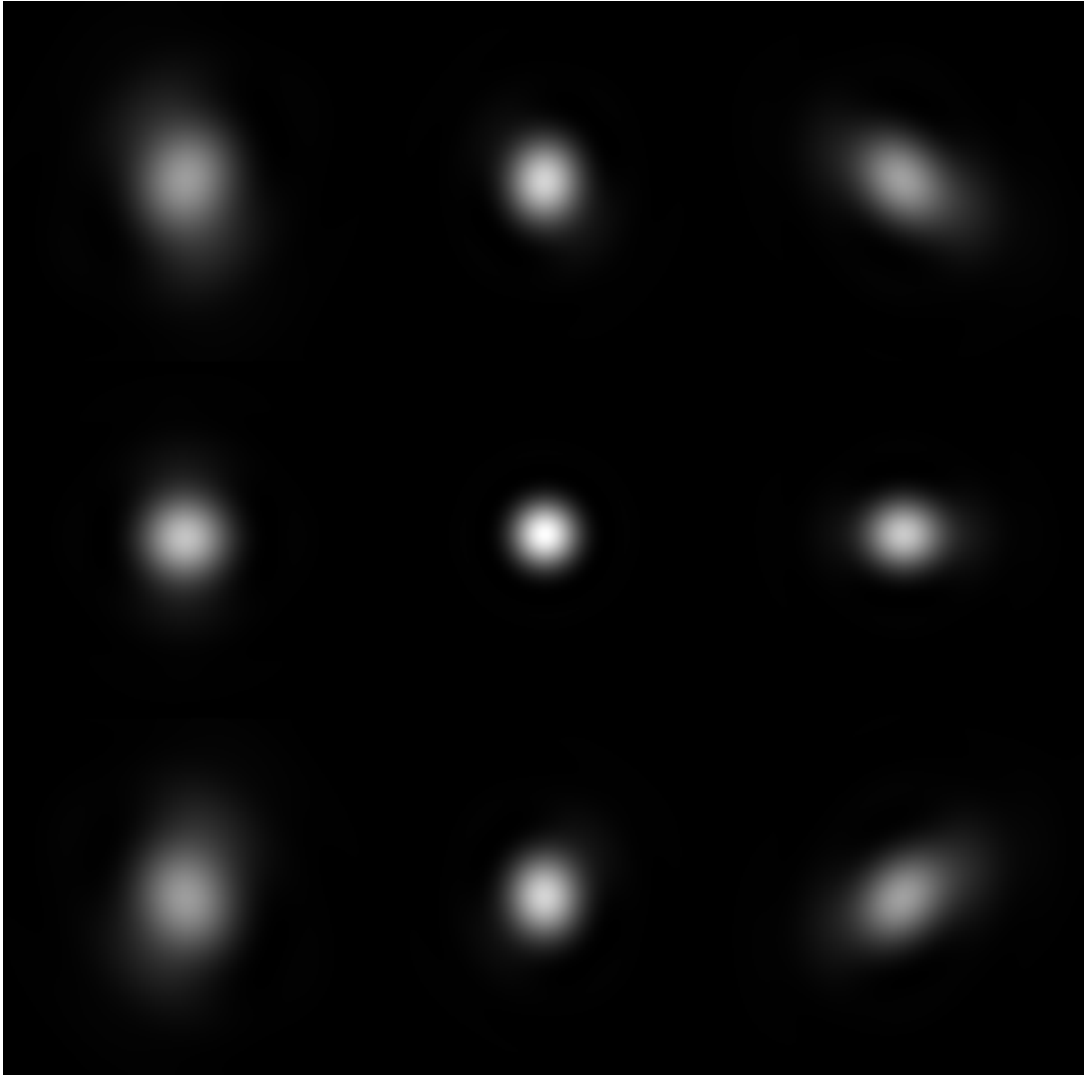


Figure 4.5: The 340 GHz active imaging system's PSFs

The PSF gives best resolution in centre of the field of view as expected. Near the ends once can see the PSF has spherical aberration. This is because focusing parabolic mirror could not be optimized for two vibrating Galvo mirrors. All the results shown in chapter 5 are calculated only using these 9 PSFs and their interpolations.

Figure 4.6 shows the MTF curve calculated using the Huygens principle for the centre of field of view. This MTF curve was used to determine the pixel number that will be used in this optical system. The resolution is calculated using the percent 50 MTF which yields 0.0337 cycles per millimeter. Multiplicative inverse of this number is the minimum separation required to distinguish a line pair. Therefore half of this number gives the resolution. Calculation yields the resolution for this system to be $\frac{1mm}{(2)(0.0337)} = 15mm$. Since the field of view is 50 cm x 50 cm, this corresponds to approximately 30 pixels x 30 pixels.

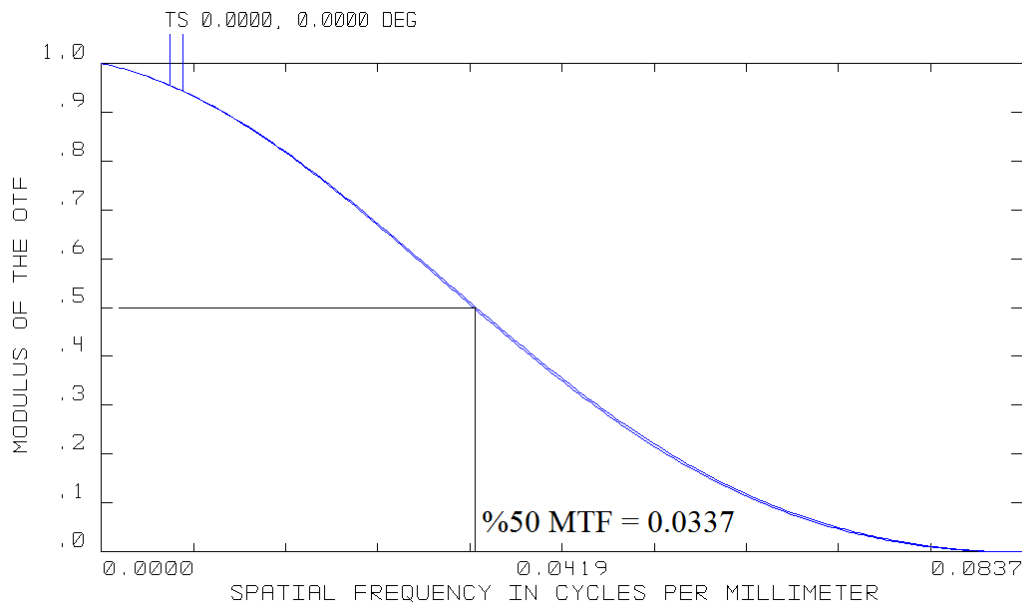


Figure 4.6: The 340 GHz active imaging system’s central MTF

4.2 COMPRESSIVE SENSING IMAGING SYSTEM

The CS imaging system used to study encoding and decoding used a single receiver position. Therefore it is of similar configuration to figure 1.3. The exact configuration is shown in 4.7. The transmitter used in this setup is similar to the 340 GHz transmitter discussed before. Only the multipliers are different; resulting in a 120 GHz frequency output instead of 340 GHz. The lenses used were Teflon. The receiver is a Golay cell (Tydex TC-1T), which was explained in previous chapters. The Golay cell, as discussed before, measures the change in

intensity of the signal. Therefore the transmitter was amplitude modulated. The modulation was controlled using the waveform generator. Finally the amplitude of the received signal was measured using a lock-in amplifier referenced to the waveform generator.

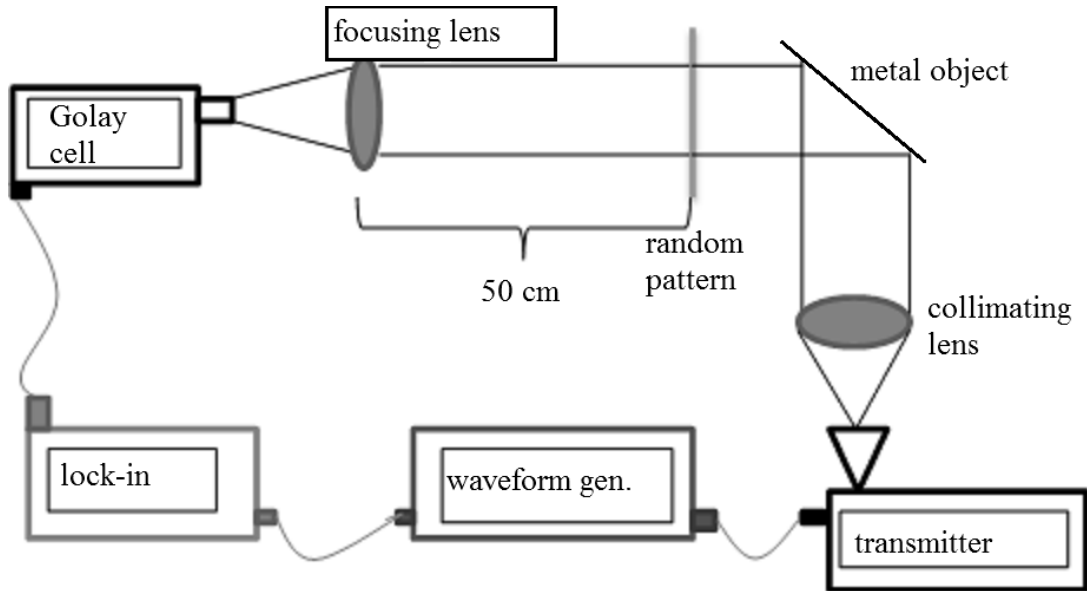


Figure 4.7: The CS imaging system

As mentioned in the previous chapter to reconstruct the image the mm-wave intensity has to be patterned randomly over the object. One of the random patterns used in measurements is shown in figure 4.9. The patterns used in the imaging system were produced using pulsed laser source (Fiberlast NanoMark 20W). The laser spot size on the 200 micron thick aluminium sheet was $30\mu\text{m}$. The patterns were designed using a code developed in Matlab (See Appendix A). All the random patterns had exact 50 square holes, out of possible 100. The holes were of size $3.0\text{mm} \times 3.0\text{mm}$. The grids between holes were $200\mu\text{m}$ wide.

To investigate clearly the development and effect of the reconstruction algorithms on the object, we had to choose an object which was not only three dimensionally flat but also had a distinctive shape with both straight and curved features in order to assess the resolution in the image reconstruction. The target (reverse letter R) was also produced with the same laser, again using an aluminium sheet. The target as seen from the receiver's point of view is given

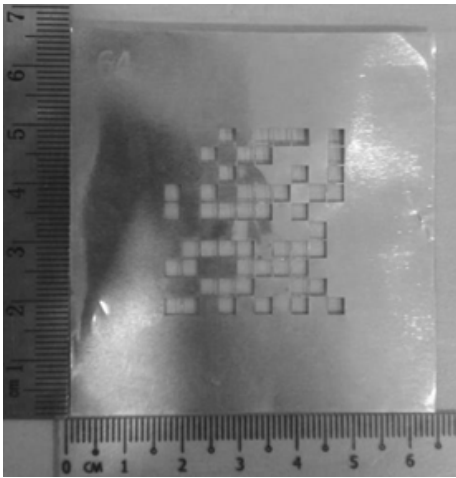


Figure 4.8: One of the random patterns used

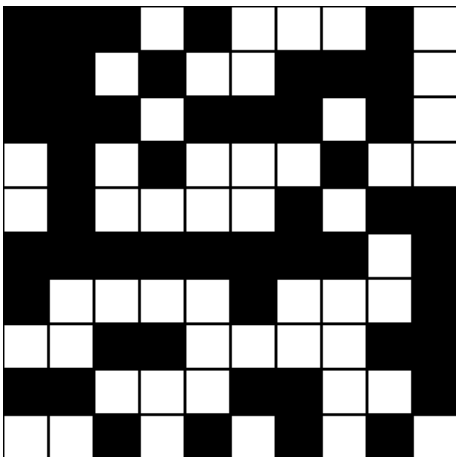


Figure 4.9: The same random pattern shown in 4.9 is shown here as it was designed

in figure 4.10.

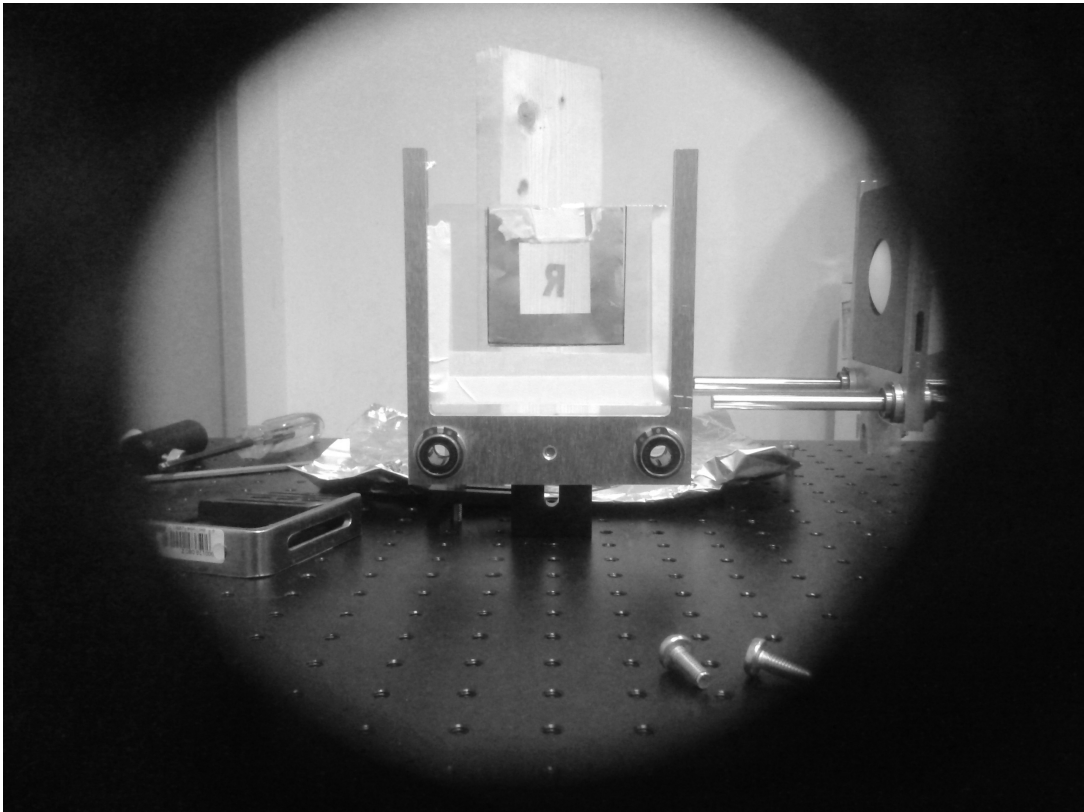


Figure 4.10: The target as seen from the receiver's point of view

The distance between the random pattern and the receiver was fixed at 50cm to ensure that any higher order diffracted beams would not be in the field of the Golay Cell detector and lens assembly. To date, and to the best of the author's knowledge, no CS experiments have been performed in reflection mode for this frequency where the random pattern was kept at a distance from the flat object. Typically, the random pattern is kept close to the target to minimize the non-specular scattered field along the optical axis towards the detector. This turned out to be a problem during the reconstruction. Another issue was the fact that while the entire optical system was designed with a He Ne laser beam to ensure that all optical components lie along the optical axis, the mm-wave emission from the horn antenna had to be placed for best possible signal transmission into the receiver. For this reason the focusing lens in front of the receiver had to be shifted slightly from the optical axis to increase signal

reception. As the results will show this caused a slight shift of the object from the centre of the reconstructed field which we compensated for in the analysis. Detailed information on imaging performance and obtained images will be given in chapter 5.

4.3 THz PASSIVE IMAGING SYSTEM

As discussed before the image processing algorithms were investigated using a third imaging system: a THz passive imaging system. Due to our groups collaboration with researchers from the Military University of Technology in Warsaw, Poland we had access to use a commercially available passive THz imaging system and aid in the development of image reconstruction algorithms which was the goal of the research group there, led by Dr. Norbert Palka.

<http://www.optolab.wat.edu.pl/index.php/kontakt/koordynatorzy-zadan/6-norbert-palka>

Basically, the purpose was to be able to modify and present the raw data collected by the imaging system as a usable mm-wave image. A model of the optics of the imaging system is shown in figure 4.11

The vibrating mirror can vibrate in two orthogonal directions, therefore a single vibrating mirror suffices to obtain a 2D image. The target is generally too far away from the system, therefore the signal can be assumed to be falling parallel on the vibrating mirror. The off-axis parabolic mirror focuses the received signal on the receiver. The receiver is similar to the receiver used in the 340 GHz system; the only difference is in multipliers. The operating frequency of the passive system is 250 GHz. To understand the image reconstruction process and effects of resolution for dealing with such a large wavelength imaging system we obtained raw data and optimized them using our own developed algorithms and finally compared our results to that which the system did by itself.

Effects of image processing on raw data acquired using the passive imaging system are discussed in detail in chapter 5.

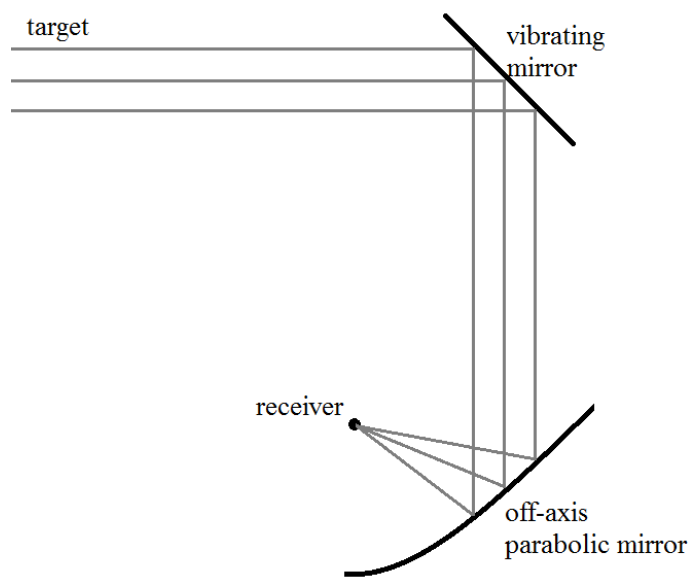


Figure 4.11: Optics of the passive imaging system

CHAPTER 5

RESULTS

Before presenting the results it is appropriate to list what experiments have been done and what simulations were completed. As mentioned in the previous chapter this thesis covers three different imaging systems.

The first system, which is the main focus of this work, is the 340 GHz system which is yet to be completed. Therefore no experimental result is presented for the 340 GHz system. However the optical design of the system was present, therefore modelling of the system's imaging performance was investigated using PSFs.

The second system is the CS system, which was used to investigate encoding and decoding experimentally. In addition to experimental results, the CS algorithms were also run with simulated data. This was done in order to better understand the imaging performance.

Finally the third system, the THz passive imaging system, was used to investigate effects of image processing algorithms. The system was also available and the experimental results will be shown. However since the system was not produced but purchased a detailed modelling of the system was not possible. Therefore only experimental results are listed.

In table 5.1 the availability of experimental and modelling/simulation results for different imaging systems are listed.

Since for CS both types of results are available, the first two sections focus on

Table5.1: Available results

Imaging system	Experimental results	Simulation/modelling results
340 GHz active imaging system	not available	available
CS imaging system	available	available
Passive imaging system	available	not available

the CS imaging system. The third section focuses on the passive imaging system and the final section covers the 340 GHz system.

5.1 CS IMAGING SYSTEM SIMULATIONS

In order to understand the image reconstruction algorithms' performances at different noise levels, the encoding procedure is simulated. The simulation was done based on the photo of the target from the receiver's point of view, shown on figure 4.10. Using the target and the random patterns the ideal signals to be received was calculated for every different random pattern.

The decoding process was completed without any alterations. So the algorithms were exactly as given previously in this work. The ideal data itself was found to be insufficient to characterize the performances of the algorithms. Therefore the ideal data was added percentile random noise, to simulate noisy measurements. The added percentiles were: 5 percent and 10 percent. In addition, a pure noise was also used as measurement. The motivation for this was to understand how the algorithms respond to random noise.

As mentioned before, the CS algorithm does not require 100 measurements to produce a 100 pixel image. Therefore the effect of increasing the number of measurements on imaging performance was also of interest. To this end the imaging performance was analysed for any number of measurements between 20 and 98. The last two random pattern square matrices were not produced due to issues with the fiber laser cutting system. However, the 98 generated patterns were enough in understanding the limits of this technique. At every different measurement a different random pattern was used.

The ideal image that was used to simulate the encoding is shown in figure 5.1. The ideal image is discretized to 10 pixel x 10 pixel for convenience. Since the random patterns were of 100 possible holes, arranged in 10 x 10, the best possible resolution one could get from the imaging setup is 10 pixel x 10 pixel. See Appendix B for the code used to discretize the target object. Due to the shift of the 120GHz light field from the optical axis the simulated reverse letter R was shifted 2 pixels to the left and one pixel to the top in order to ensure accurate comparison with experimental results which are treated in the next section. The goal of this work is to compare the simulated best possible reverse R that can be obtained using a 10 x 10 square array with a 3mm pixel size to the actual image recorded at 120GHz with the same array using CS techniques. Immediately one can expect that the 120GHz will not be as good as this image due to the large wavelength (near 2.1mm), however this analysis still demonstrates the limitations and advantages of the CS technique.

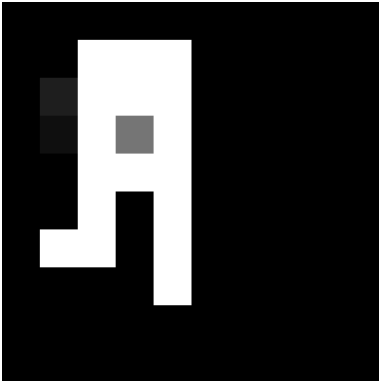


Figure 5.1: Shifted ideal image

Before moving on to show the imaging performance, it is appropriate to define a metric to measure imaging performance. Various methods can be used to evaluate imaging performance, In this work Structural Similarity (SSIM) is used. SSIM is a metric used to compare similarity between two images [39]. Assume x and y are two images of same resolution, then SSIM of them is given by:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Here μ_x is the average of x, μ_y is the average of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y and finally σ_{xy} is the covariance of x and y. c_1 and c_2 are used to stabilize divisions with weak denominator. They are calculated using the possible contrast between two pixels(K): $c_1 = (K/100)^2$ and $c_2 = (3K/100)^2$. These are conventional constants for SSIM. In our simulation this contrast was chosen to be 255, therefore $c_1 = 6.5025$ and $c_2 = 58.5225$ was used in calculations. The SSIM has a maximum level of 1, implying two images are exactly same and a minimum level of -1, implying the two images are as dissimilar as possible. For two random images the SSIM is expected to be around zero.

By definition SSIM compares two images. Obviously the first image to be compared is the reference image given in figure 5.1. The other image is the reconstructed image. The reconstructed image is reconstructed using a variable number of measurements, between 20 and 98 with either l_1 minimization algorithm or TV minimization algorithm. Those algorithms are defined in chapter 2 section 6.

In all the figures in this subsection, a solid line is the ideal data without any noise, the line marked with circles (o) is ideal data with 5 percent noise added, the line marked with plus signs (+) is ideal data with 10 percent noise added and finally the line marked with asterisk (*) is the pure noise. In figure 5.2 the evolution of SSIM with increasing number of measurements is shown for l_1 reconstruction algorithm. In figure 5.3 the evolution of SSIM with increasing number of measurements is shown for TV reconstruction algorithm.

As can be seen in figures 5.2, 5.3 both algorithms are very successful for ideal data, and decrease in performance with increasing levels of noise. The image reconstructed using pure noise has an average SSIM of almost 0 as expected. It is important to note that even for the ideal data at least 38 measurements must be used to construct the image properly. As can be seen from graphs, with increasing noise levels the minimum number of measurements required is increasing. No noteworthy difference in performances of two different algorithms was found. Finally it is important to note that the SSIM of the ideal data with 5 percent noise added is not stable. As can be seen it fluctuates by 0.2. This

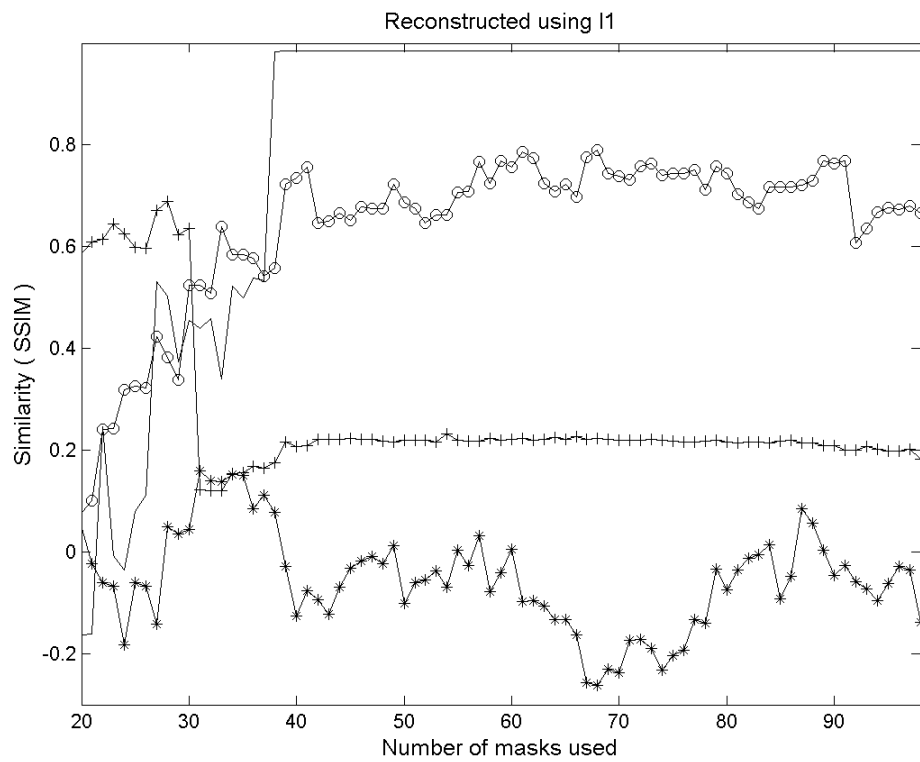


Figure 5.2: Evolution of SSIM with increasing number of measurements, using l1 reconstruction algorithm. (-) ideal data, (o) ideal data + 5 percent noise, (+) ideal data + 10 percent noise, (*) pure noise

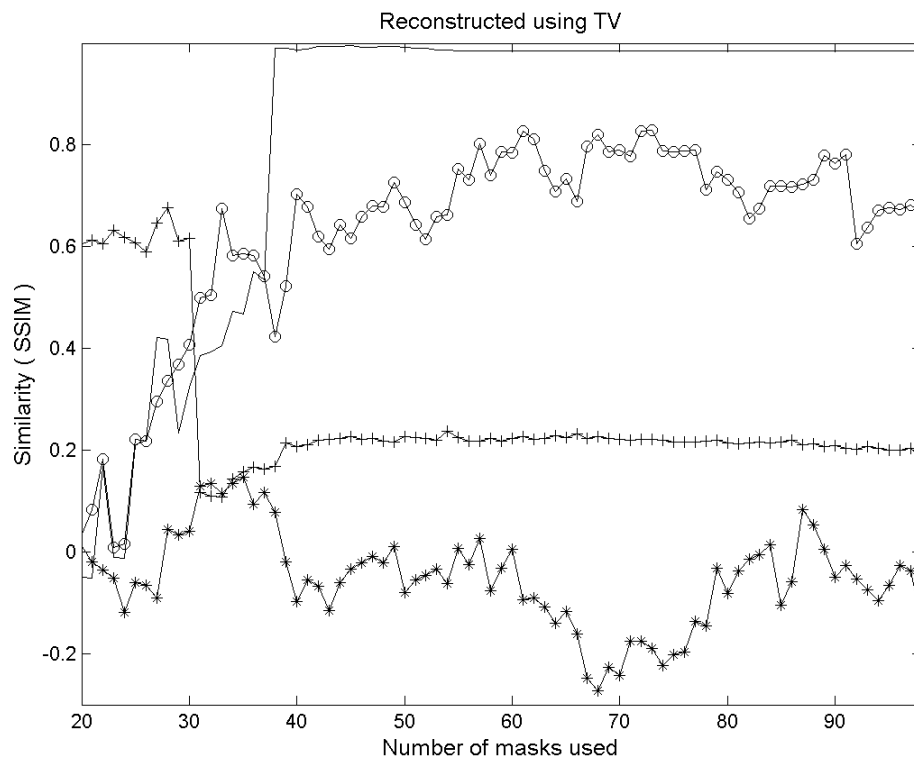


Figure 5.3: Evolution of SSIM with increasing number of measurements, using TV reconstruction algorithm. (-) ideal data, (o) ideal data + 5 percent noise, (+) ideal data + 10 percent noise, (*) pure noise

can be because the random noise added to the system has a positive impact on image reconstruction at some times, and negative in some other.

The reconstructed images using 38 measurements and 98 measurements are shown in figure 5.4. As can be seen, if there is noise then the image is not formed properly at measurement number 38, although when there is no noise it does. Finally it is important to note that for pure noise, increasing number of measurements complicate the image randomly.

5.2 CS IMAGING SYSTEM EXPERIMENTAL RESULTS

In this section the results for measurements taken in experiment are listed. The measurements were taken one by one: The first random pattern was placed and the received signal was measured, then the second and so on. . . In addition to this measurement an additional measurement was done, in order to eliminate the background signal. For this second measurement the target (reverse R) was removed from the system and measurements were taken for every matrix again. Therefore the background, independent of the target, was recorded. In reconstruction the ratio of these two measurements were used as input. This ratio of signals for all 98 patterns were combined to form the image, using the CS reconstruction algorithms described in chapter 2, section 7. Details of the code developed for this task can be found in Appendix C. The final images, using all 98 measurements, are shown in figure 5.5 and 5.6.

The SSIM is computed again with comparing reconstructed images with the reference image shown in 5.1. For better analysis the SSIM graphs are plotted not only for the experimental results. The ideal data and pure noise graphs from the previous chapters are also included in plots.

In all the figures in this subsection, a solid line is the ideal data without any noise, the line marked with circles (o) is the data obtained with the experiment and finally the line marked with asterisk (*) is the pure noise. In figure 5.7 the evolution of SSIM with increasing number of measurements is shown for $l1$ reconstruction algorithm. In figure 5.8 the evolution of SSIM with increasing

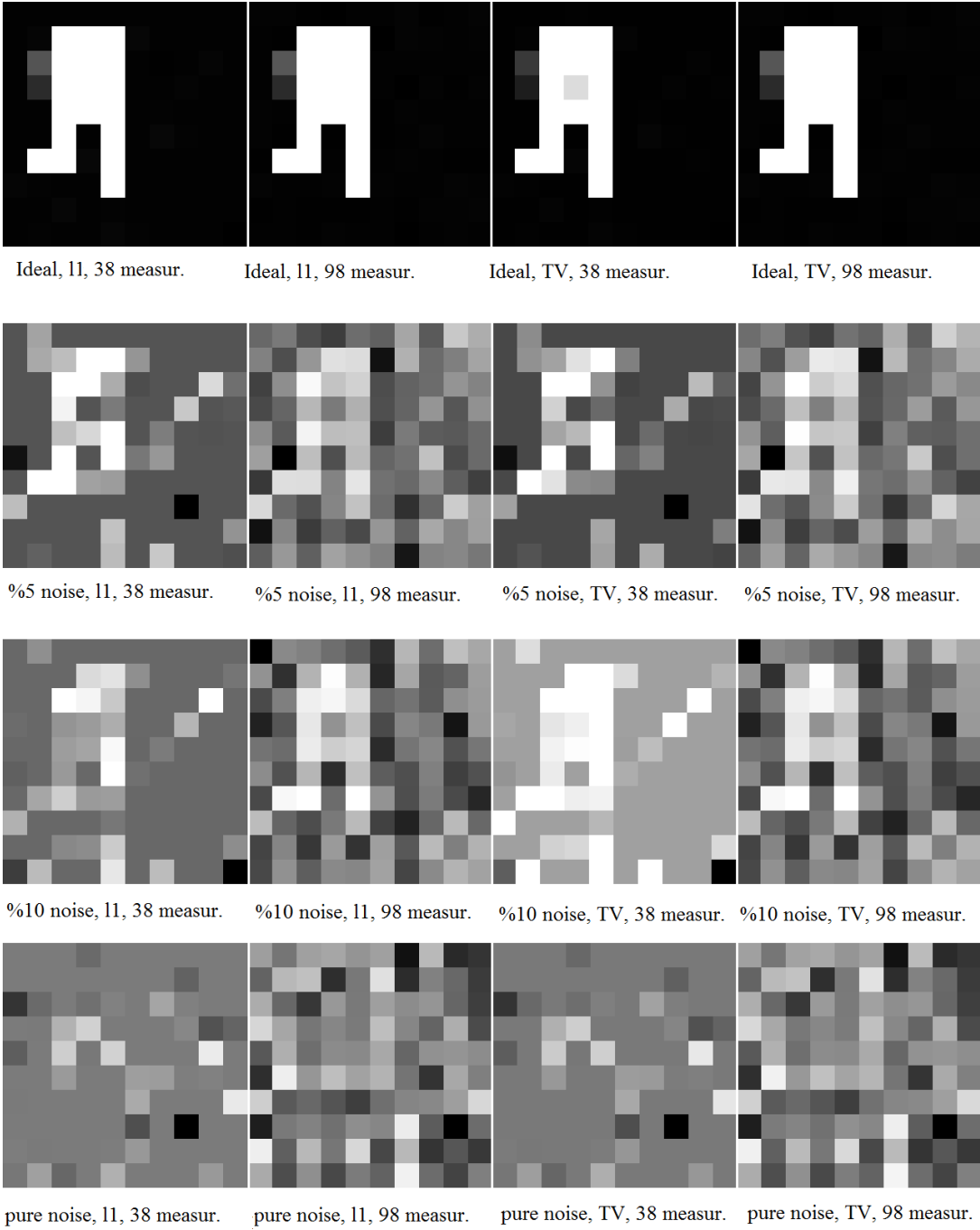


Figure 5.4: The reconstructed images using 38 measurements and 98 measurements

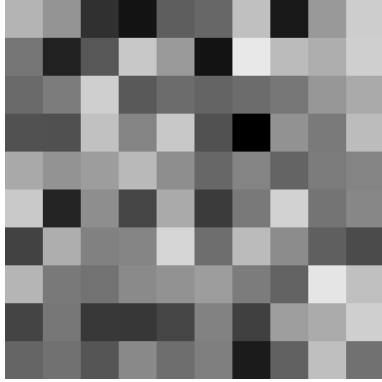


Figure 5.5: Final image obtained using experimental data using l1 algorithm

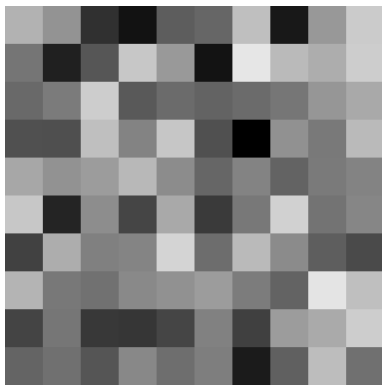


Figure 5.6: Final image obtained using experimental data using TV algorithm

number of measurements is shown for TV reconstruction algorithm.

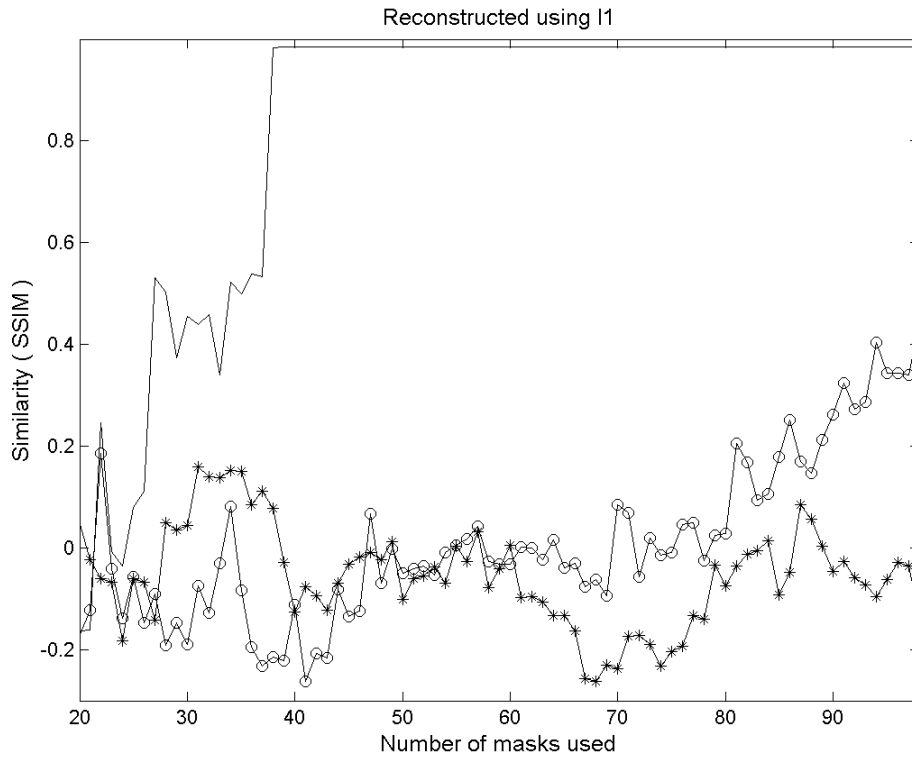


Figure 5.7: Evolution of SSIM with increasing number of measurements, using $l1$ reconstruction algorithm. (-) ideal data, (o) experimental data, (*) pure noise

As can be seen the performances of the algorithms do not differ for any of the inputs. Considering the simulations with ideal data with noises, it is normal that the experimental data forms an image much later than the ideal data. Except from late formation of image, the imaging system successfully reconstructs the image, the similarity is greater than 0.4 after 98 measurements are considered. The similarity increases in last 20 measurements rapidly. This may be explained by noise in the acquired data. In the simulations discussed in the previous section noise was distributed evenly on every measurement, however this is not necessarily valid for our system. Therefore image formation may have been shifted.

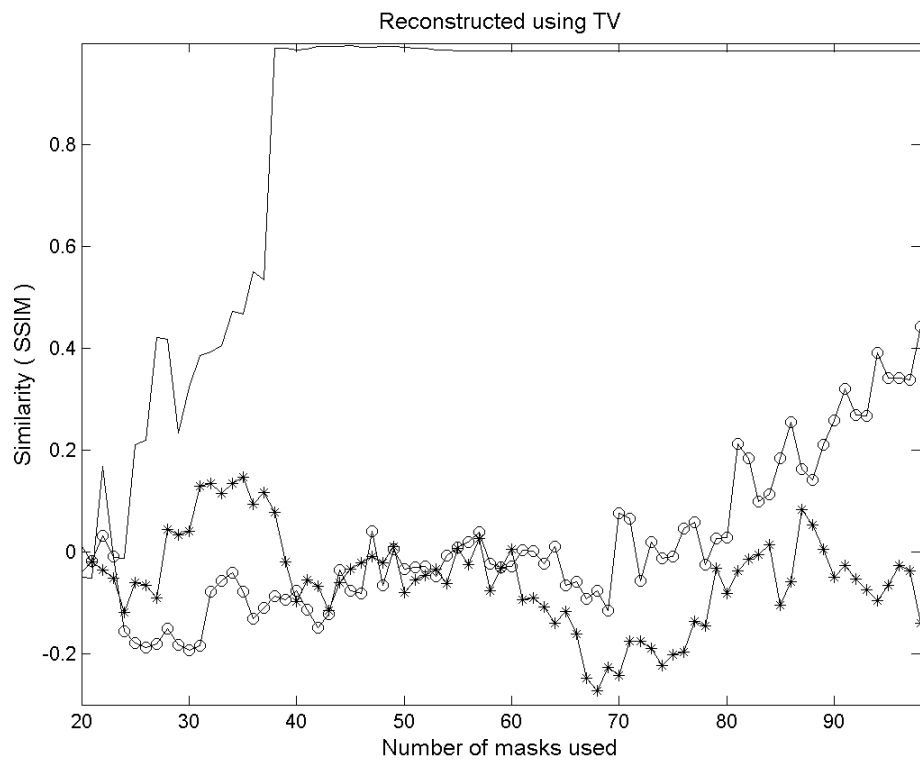


Figure 5.8: Evolution of SSIM with increasing number of measurements, using TV reconstruction algorithm. (-) ideal data, (o) experimental data, (*) pure noise

5.3 PASSIVE IMAGING SYSTEM EXPERIMENTAL RESULTS

The passive imaging system was investigated in order to understand the effects of image processing algorithms. The goal of this work was to show that algorithms applied to the raw data by us could give similar results as to what is obtained using the processing already developed in the commercial passive imaging system tested here. This capability would allow the user to control and manipulate the passive image as he or she deemed fit. For example, it was the goal of the research group in Poland to fuse the passive image with a visible image giving the operator a better understanding of threat objects.

To test the passive imaging system performance the team in Poland designed a mannequin that was able to simulate the human body. By covering the "dummy" with tubes they were able to pass water at the appropriate temperature to simulate a human being under light clothing. During this task they discovered that the commercial system does not give the raw data as is observed in the final image. Instead the system mixes the data in a seemingly random pattern before it processes it. This is thought to ensure that 3rd parties do not manipulate the system or its control software without the developers' consent. In order to work around this drawback we designed a scenario where a hot water filled tube that made an S shape was imaged. The seemingly random raw data was analysed to understand the reconstruction. The result of the reconstruction is shown in Figure 5.9, and the code written in Matlab used for this task is given in Appendix D and E.

In the next step our goal was to see how one can manipulate the raw image to show an image comparable to what the commercial system already does with its own proprietary algorithms. This approach is outlined as follows: First image obtained by commercial system and target (which is a knife placed on the mannequin) will be analysed. Then, the reconstructed raw image will be processed by first applying an averaging algorithm followed by an edge detection algorithm and a thresholding algorithm, details of which were given in Chapter 1, section 3. Finally the obtained image will be compared to that already obtained with the system.

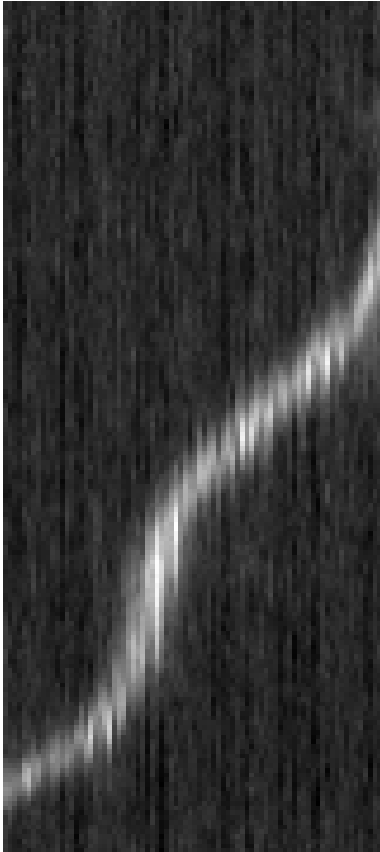


Figure 5.9: Reconstructed image

5.3.1 Target

The target used in this system is shown in figure 5.10. It is a water-heat pumped mannequin. The threat object placed on the mannequin is a knife. The knife is shown in figure 5.11

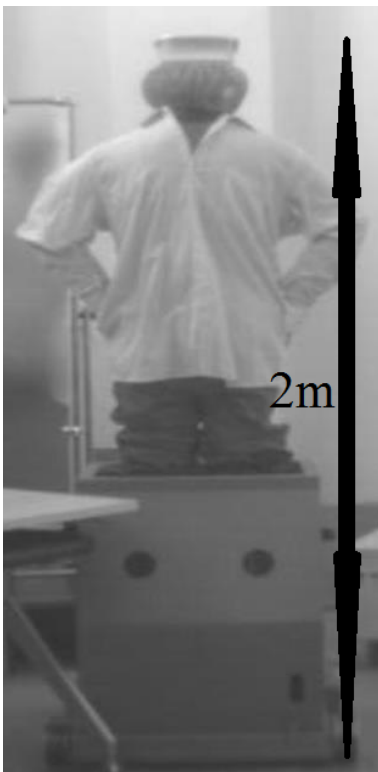


Figure 5.10: The target mannequin

The image obtained using the built-in algorithm in the passive THz imaging system is shown in figure 5.12. The raw data obtained from the system is shown in figure 5.13.

As can be seen the raw image is processed very successfully by the passive imaging system's built-in algorithm. In the following subsections an alternative processing will be investigated.

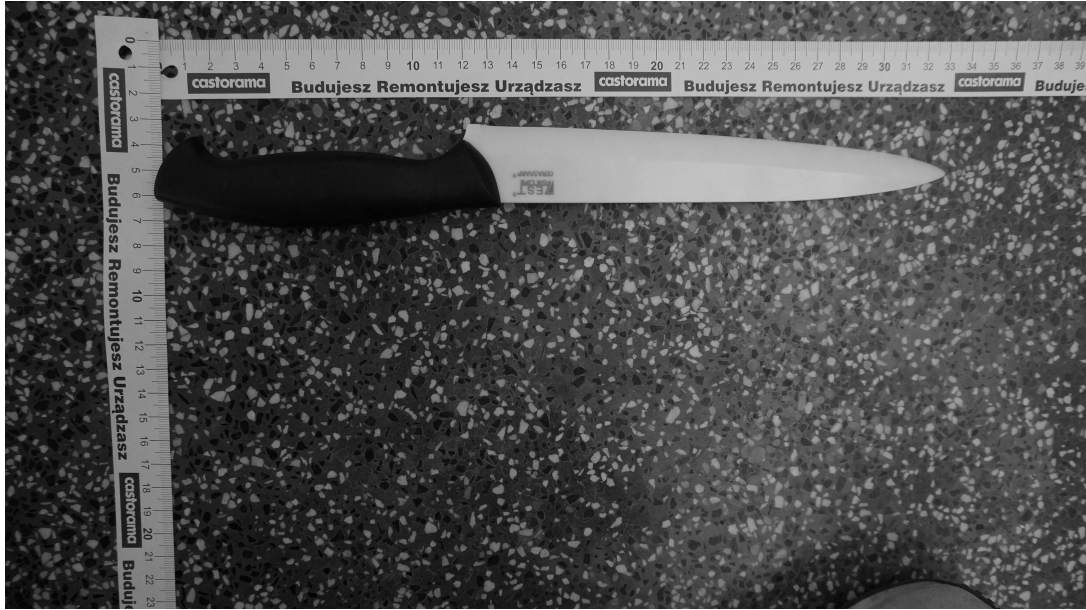


Figure 5.11: The threat object



Figure 5.12: Image obtained from built-in algorithm of the passive imaging system, resolution is 271 x 124 pixels

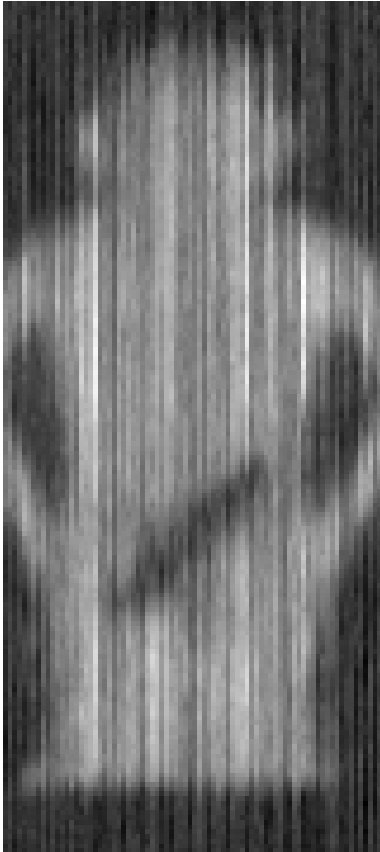


Figure 5.13: Raw image obtained from the passive imaging system

5.3.2 Averaging

The averaging algorithm used in this work is Gaussian averaging. Every pixel is averaged using the Gaussian weighed sum of its neighbouring pixels. The code for this is given in Appendix F. The image obtained after this averaging is shown in figure 5.14.

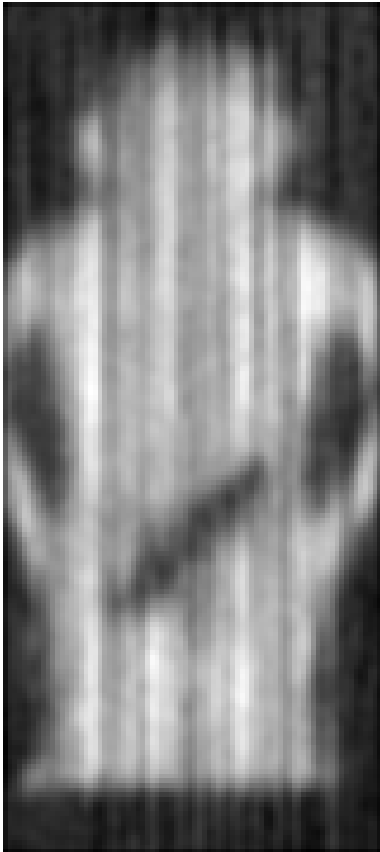


Figure 5.14: Averaged image

As can be seen in figure 5.14 the averaging blurs the image in a positive way. After averaging the image becomes smoother. The reason why raw data is not smooth can be explained using the vibrations of the vibrating mirror of this system. The vibrating mirror completes a whole vertical line before moving on to the next horizontal line, resulting in a horizontal discontinuity in the obtained image. The image obtained from the system's built-in algorithm can be used as a reference for an SSIM comparison of both the raw image and the image obtained after averaging. The SSIM for the raw image is found to be 0.2271

while it became 0.3438 after the averaging.

5.3.3 Edge detection

Nonlocal means (NLM) algorithm is a nonlocal averaging algorithm which can be used for edge detection [40]. The NLM does not average a pixel based on its neighbourhood. Instead it finds the most similar pixels to it in the whole image. Then it uses this pixel to smooth out the pixel at hand. Therefore using nonlocal information to smooth a pixel, the algorithm has the ability to reveal edges more clearly. The image obtained after applying a nonlocal means algorithm is shown in figure 5.15.

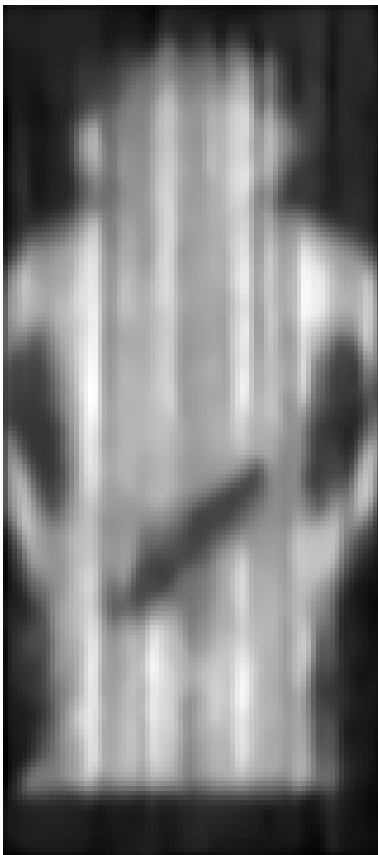


Figure 5.15: Image after NLM algorithm

The SSIM after NLM algorithm is applied is found to be 0.3822. Clearly the NLM algorithm is most useful near the edges. This is natural since the algorithm

uses nonlocal pixels to correct the pixels at the edges.

5.3.4 Thresholding

Thresholding is the final part of the image processing algorithms that will be discussed in this work. In general it is customary to finalize the process with an image fusion. Although the image fusion is being investigated the results are not yet ready, therefore no work is shown in this work.

Thresholding is done using the average of the whole image in this work. Although there are nonlocal thresholding algorithms for more precise edge detection, a global thresholding was decided to suffice for the work in Poland. The image obtained after applying a thresholding algorithm is shown in figure 5.16.



Figure 5.16: Image after thresholding

The SSIM obtained after the thresholding algorithm is applied is 0.4095. Clearly

the image after thresholding is much more similar to image processed by system's built in algorithm compared to the raw image obtained.

Therefore one can say that an imaging systems' performance can be increased greatly (by almost a factor of 2 in case of SSIM) by post-processing. In the 340 GHz active imaging system which is currently being developed, a post-processing algorithm is being developed for better recognition of threat objects.

5.4 THz ACTIVE IMAGING SYSTEM SIMULATION RESULTS

To model the imaging performance of the system the Huygens PSF, explained in chapter 3 section 1, was used.

To simulate the imaging system the field of view has to be divided in to the number of pixels that is dependent on the resolution of the active imaging system. For the 50 x 50cm field of view, this turned out to be 30 x 30 pixels. This means that ideally we would need 900 PSFs to accurately simulate the target object with our active imaging system. However, due to the similarity of the PSFs we only used the PSFs in the edges and the centre of the field of view and interpolated the values in between using the following formula:

$$PSF(x, y) = \frac{PSF_1(a, b)}{\sqrt{(a-x)^2 + (b-y)^2}} + \frac{PSF_2(c, d)}{\sqrt{(c-x)^2 + (d-y)^2}} + \frac{PSF_3(e, f)}{\sqrt{(e-x)^2 + (f-y)^2}}$$

where $PSF(x, y)$ is the PSF used in calculating the current pixel; PSF_1 , PSF_2 and PSF_3 are the three closest points of the 9 used in interpolation. The imaging was simulated by defining a target and imaging it via convolutions of PSFs. The interpolation algorithm and the algorithm for taking the mentioned convolution can be found in Appendix G.

5.4.1 Noise-free simulations

In this subsection ideal images are distorted only with PSF of the optical system. No effect from measurement errors is considered.

In figure 5.17, the first investigated target is shown. Since the field of view of the imaging system is approximately 50 cm x 50 cm, one can assume that the target is two metal rods of width 10 cm placed vertically. The resulting image is shown in figure 5.18. As seen, the PSF of the imaging system causes a blurring in the lines.

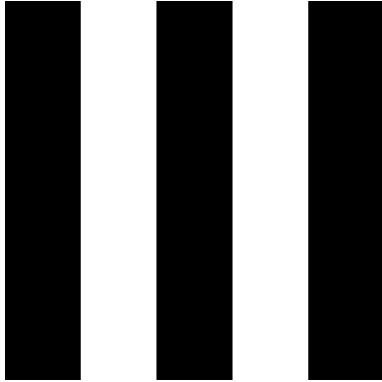


Figure 5.17: First target

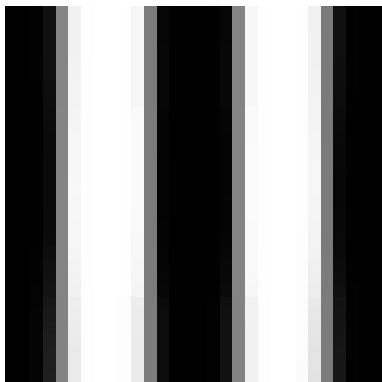


Figure 5.18: Image obtained from first target

In figure 5.19, the second investigated target is shown. One can assume that the target is two metal rods of width 10 cm placed horizontally. The resulting image is shown in figure 5.20. As seen, the PSF of the imaging system causes a blurring in the lines. In order to emphasize the imaging performance a third target was used. This time the target can be thought as squares of 5cm width. The target is shown in figure 5.21 and the image is shown in figure 5.22. Clearly imaging performance is best at the middle of the field of view and it gets worse near the ends.



Figure 5.19: Second target

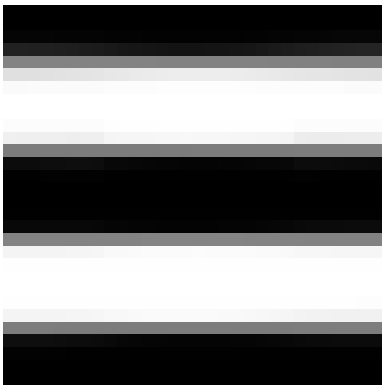


Figure 5.20: Image obtained from second target

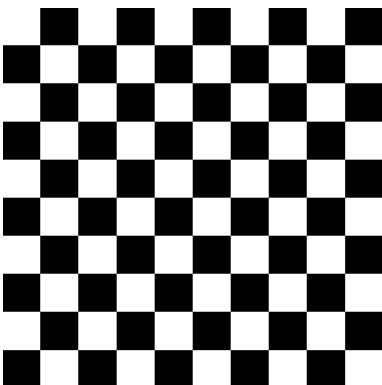


Figure 5.21: Third target

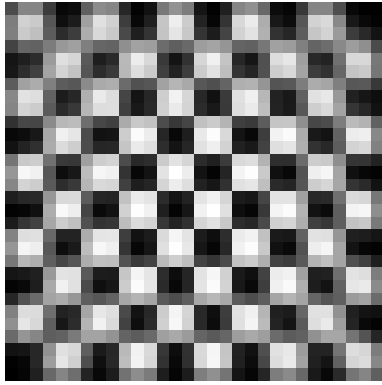


Figure 5.22: Image obtained from third target

The targets discussed to this point were only to analyse the performance of the imager. The final target is a porcelain knife carried in pocket. The signal levels were investigated in Poland, using a reflection based TDS system. The target is shown in figure 5.23 and the obtained image is shown in figure 5.24.

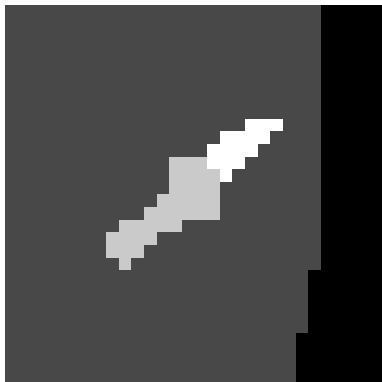


Figure 5.23: Porcelain knife

5.4.2 Simulations with random noise

Since the source of random noise has an unpredictable nature, as discussed before in chapter 2 section 5 and also in chapter 3 section 2, the random noise is added to the images as percentile noise and is allowed to be both positive and negative. All noise related work is done on the knife shown in figure 5.23. Different levels of noise are added to the system and its effects are measured using SSIM. This

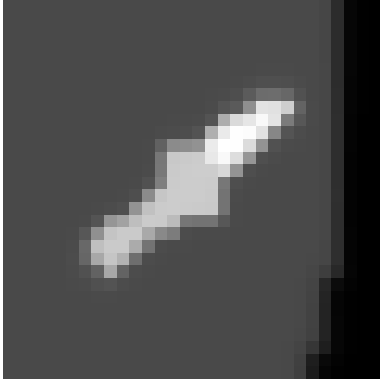


Figure 5.24: Image obtained from porcelain knife

random noise can be thought as temperature fluctuations in the field of view of the system.

Figure 5.25 shows the image obtained with 10 percent noise. Figure 5.26 shows the image obtained with 20 percent noise. Figure 5.27 shows the image obtained with 40 percent noise. As can be seen from figures, even with

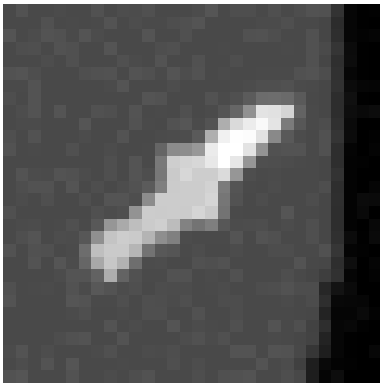


Figure 5.25: Image obtained from porcelain knife with 10 percent noise

5.4.3 Effects of the receiver integration time

As discussed before, in chapter 2 section 5 and also in chapter 3 section 2, the only systematic noise that is investigated in this work is the noise caused by long integration time in receiver.

To calculate the ideal integration time required from the receiver in this system,

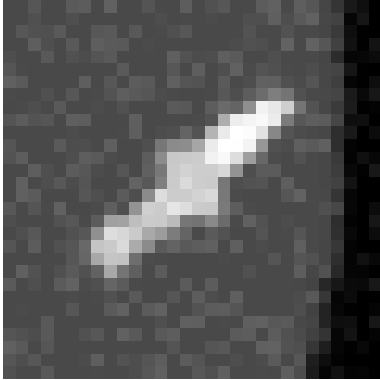


Figure 5.26: Image obtained from porcelain knife with 20 percent noise

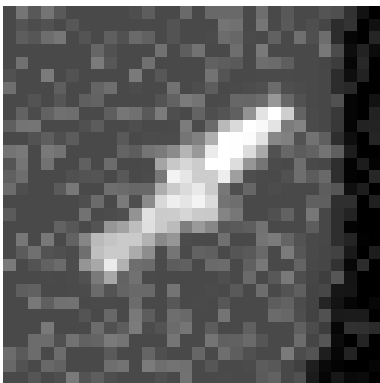


Figure 5.27: Image obtained from porcelain knife with 40 percent noise

one needs to find how much time the system spends on each pixel. In the developed imaging system, the refresh rate is found to be 2 frames per second. Since every frame has 900 pixels, a pixel is measured in $1/1800$ seconds which corresponds to roughly 0.56 milliseconds. Assume the integration time of the receiver is greater than that time. Then the image will be distorted accordingly.

This distortion is modelled by adding the signals from previous pixels to reconstruct the next pixel. If one considers the distortion can be modelled exponentially, then the resulting image for a decay constant of 30 pixels (16ms) is shown in figure 5.28. Another image for a decay constant of just 3 pixels (1.6ms) is shown in figure 5.29. Appendix H gives the algorithm used to calculate these images.



Figure 5.28: Image obtained with integration time = 30 x ideal

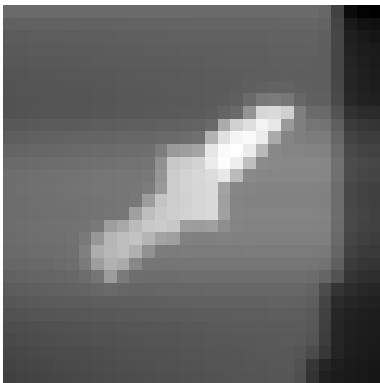


Figure 5.29: Image obtained with integration time = 3 x ideal

In simulating this noise, the Galvo mirrors are assumed to complete a horizontal line from left to right first. Then they are assumed to move on to the next line,

again to start from left and move to the right. As can be seen from figure 5.28 and 5.29, the response time of the receiver is crucial in image reconstruction. If the receiver is unable to respond to changes in less than 0.56 milliseconds, we expect distortions in the image obtained from the system.

CHAPTER 6

CONCLUSION

In this work, 3 different imaging systems were analysed; compressive sensing imaging system, passive imaging system and active imaging system. The goal of the CS imaging system was to image an object below the Nyquist rate using a 120 GHz active system non-scanning system. The used masks were of 3mm pixel size and had a 10 x 10 array geometry. The goal of the passive imaging system was to successfully reconstruct raw data obtained from a commercially available THz passive imaging system, then using algorithms to process the image and compare this processed image with the one obtained from the commercial system's built in function. The goal of the active imaging system was to simulate response of the receiver and optical system to a perfect target scanned by a source of 340 GHz.

First the CS imaging system was analysed to investigate image encoding and decoding experimentally. For this system two different reconstruction algorithms, l_1 minimization and TV minimization, were implemented with success. Moreover simulations were completed to investigate algorithms' and experiment's success. The results show that image acquisition and reconstruction was completed with partial success. Although image was formed, it required more patterns to form compared to expectations. This may be because of the fact that the target and the random pattern were not situated as closely as possible, therefore allowing background's specular reflection to be received and processed in image reconstruction. Another possible explanation for low image quality is the metric that is used. In evaluating the imaging performance the reference image was a shifted

image. The shifting was used for integer pixel shifts. It may be that the actual shifting was not in integer numbers, hence the reference image we compare our reconstructed images may be wrong. Therefore resulting in lower SSIM values than reality.

The second system analysed was the passive imaging system. This system was used to test three image processing algorithms; averaging, edge detection and thresholding experimentally. Experimental results show that all three algorithms are helpful in identification of threat objects. Although the algorithms were found to be successful, one may argue that the reference image used in measuring their performance using SSIM may be wrong. Since SSIM was calculated using the image obtained from the imaging system as a reference, one may argue that if the built-in algorithm was not optimal, then the comparison was not reliable. Therefore to increase confidence in the developed algorithms one may need to test them using a better analysed optical system.

The third system analysed was the 340 GHz active imaging system which is currently being developed in Middle East Technical University. In modelling this system PSFs were interpolated and using convolution the imaging performance was investigated. The imaging performance's robustness to noise was investigated using two different types of noise. Perfect targets were simulated at the focus of this imaging system, which was 5m from the elliptical mirror. The results show that there is significant distortion at the edges of the field of view. This confirms that scanning both X and Y directions using vibrating mirrors cause significant aberrations at the edges.

Future work entails that the simulations will be compared with experimental results. The operation of this active imaging system can be optimized using the algorithms developed for passive imaging in this thesis.

REFERENCES

- [1] D.L. Woolard, E.R. Brown, Michael Pepper, and M. Kemp. Terahertz frequency sensing and imaging: A time of reckoning future applications? *Proceedings of the IEEE*, 93(10):1722–1743, 2005.
- [2] J.E. Bjarnason, T. L J Chan, A. W M Lee, M. A. Celis, and E.R. Brown. Millimeter-wave, terahertz, and mid-infrared transmission through common clothing. *Applied Physics Letters*, 85(4):519–521, 2004.
- [3] N. Llombart, K.B. Cooper, R.J. Dengler, T. Bryllert, and P.H. Siegel. Confocal ellipsoidal reflector system for a mechanically scanned active terahertz imager. *Antennas and Propagation, IEEE Transactions on*, 58(6):1834–1841, 2010.
- [4] P.H. Siegel. Terahertz technology. *Microwave Theory and Techniques, IEEE Transactions on*, 50(3):910–928, 2002.
- [5] Jianping Yao. Microwave photonics. *Lightwave Technology, Journal of*, 27(3):314–335, 2009.
- [6] K. Iga. Surface-emitting laser-its birth and generation of new optoelectronics field. *Selected Topics in Quantum Electronics, IEEE Journal of*, 6(6):1201–1215, 2000.
- [7] K.L. Blackard, T.S. Rappaport, and C.W. Bostian. Measurements and models of radio frequency impulsive noise for indoor wireless communications. *Selected Areas in Communications, IEEE Journal on*, 11(7):991–1001, 1993.
- [8] D.S. Odell. An optical pointer for infrared remote controllers. In *Consumer Electronics, 1995., Proceedings of International Conference on*, pages 324–325, 1995.
- [9] Bernd M. Fischer, Hanspeter Helm, and P.U. Jepsen. Chemical recognition with broadband thz spectroscopy. *Proceedings of the IEEE*, 95(8):1592–1604, 2007.
- [10] S.U. Hwu, K.B. deSilva, and C.T. Jih. Terahertz (thz) wireless systems for space applications. In *Sensors Applications Symposium (SAS), 2013 IEEE*, pages 171–175, 2013.
- [11] Heiko Neumann and Florian Raudies. Neural mechanisms for form and motion detection and integration: Biology meets machine vision. In *Proceedings of the 12th International Conference on Computer Vision - Volume Part I, ECCV'12*, pages 468–473, Berlin, Heidelberg, 2012. Springer-Verlag.

- [12] G.P. Gallerano, E. Grosse, R. Korenstein, M. Dressel, W. Mantele, M.R. Scarfi, A.-C. Cefalas, P. Taday, R.H. Clothier, and P. Jepsen. Thz-bridge: an european project for the study of the interaction of terahertz radiation with biological systems. In *Infrared and Millimeter Waves, 2004 and 12th International Conference on Terahertz Electronics, 2004. Conference Digest of the 2004 Joint 29th International Conference on*, pages 817–818, 2004.
- [13] R. Appleby and R.N. Anderton. Millimeter-wave and submillimeter-wave imaging for security and surveillance. *Proceedings of the IEEE*, 95(8):1683–1690, 2007.
- [14] R. Appleby and H.B. Wallace. Standoff detection of weapons and contraband in the 100 ghz to 1 thz region. *Antennas and Propagation, IEEE Transactions on*, 55(11):2944–2956, 2007.
- [15] P. Helisto, A. Luukanen, L. Gronberg, J.S. Penttila, H. Seppa, H. Sipola, C.R. Dietlein, and E.N. Grossman. Antenna-coupled microbolometers for passive thz direct detection imaging arrays. In *European Microwave Integrated Circuits Conference, 2006. The 1st*, pages 35–38, 2006.
- [16] E.J. Candes and M.B. Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, 2008.
- [17] T. Han, D.G. Goodenough, A. Dyk, and J. Love. Detection and correction of abnormal pixels in hyperion images. In *Geoscience and Remote Sensing Symposium, 2002. IGARSS '02. 2002 IEEE International*, volume 3, pages 1327–1330 vol.3, 2002.
- [18] D. W. Isakson. Developments in telephotography. *American Institute of Electrical Engineers, Transactions of the*, XLI:794–801, 1922.
- [19] John F. Federici, Dale Gary, B. Schulkin, Feng Huang, Hakan Altan, Robert Barat, and D. Zimdars. Terahertz imaging using an interferometric array. *Applied Physics Letters*, 83(12):2477–2479, 2003.
- [20] A. Richard Thompson, James M. Moran, and Jr. George W. Swenson. *Interferometry and Synthesis in Radio Astronomy*. Wiley, 2008.
- [21] Thomas M. Goyette, Jason C. Dickinson, Kurt J. Linden, William R. Neal, Cecil S. Joseph, William J. Gorveatt, Jerry Waldman, Robert Giles, and William E. Nixon. 1.56 terahertz 2-frames per second standoff imaging, 2008.
- [22] Ken B. Cooper, Robert J. Dengler, Nuria Llombart, Ashit Talukder, Anand V. Panangadan, Chris S. Peay, Imran Mehdi, and Peter H. Siegel. Fast high-resolution terahertz radar imaging at 25 meters, 2010.
- [23] N. Llombart, R.J. Dengler, and K.B. Cooper. Terahertz antenna system for a near-video-rate radar imager [antenna applications]. *Antennas and Propagation Magazine, IEEE*, 52(5):251–259, 2010.
- [24] J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *Image Processing, IEEE Transactions on*, 12(11):1338–1351, 2003.

- [25] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, 1986.
- [26] V. Ferrando, V. Pagnotta, and G. Pezzuto. Teraeye- a fully passive thz inspection system based on nanotechnology for security applications. In *Infrared and Millimeter Waves, 2007 and the 2007 15th International Conference on Terahertz Electronics. IRMMW-THz. Joint 32nd International Conference on*, pages 654–655, 2007.
- [27] Chris Stauffer and W. E L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages –252 Vol. 2, 1999.
- [28] S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7):674–693, 1989.
- [29] M. Kowalski, N. Palka, M. Piszczek, and M. Szustakowski. The evaluation methodology of thz-vis fused images. In *Infrared, Millimeter, and Terahertz Waves (IRMMW-THz), 2013 38th International Conference on*, pages 1–2, 2013.
- [30] K.B. Cooper, R.J. Dengler, N. Llombart, T. Bryllert, G. Chattopadhyay, I. Mehdi, and P.H. Siegel. An approach for sub-second imaging of concealed objects using terahertz (thz) radar. *Journal of Infrared, Millimeter, and Terahertz Waves*, 30(12):1297–1307, 2009.
- [31] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM J. IMAGING SCI*, pages 248–272, 2008.
- [32] Wai Lam Chan, M.L. Moravec, R.G. Baraniuk, and D.M. Mittleman. Terahertz imaging with compressed sensing and phase retrieval. In *Lasers and Electro-Optics, 2007. CLEO 2007. Conference on*, pages 1–2, 2007.
- [33] Wai Lam Chan, Kriti Charan, Dharmpal Takhar, Kevin F. Kelly, Richard G. Baraniuk, and Daniel M. Mittleman. A single-pixel terahertz imaging system based on compressed sensing. *Applied Physics Letters*, 93(12):–, 2008.
- [34] W.T. Freeman, T.R. Jones, and E.C. Pasztor. Example-based super-resolution. *Computer Graphics and Applications, IEEE*, 22(2):56–65, 2002.
- [35] Li Shengyang, Zhu Chongguang, Guo Jun, Bo Shukui, and Ye Famao. Spatial resolution improvement by iterative resampling and restoration based on mtf computation. In *Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International*, volume 5, pages 3734–3736, 2005.
- [36] M. Planck. *The Theory of Heat Radiation*. History of Modern Physics, 1800-1950. Springer-Verlag GmbH, 1906.

- [37] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1998.
- [38] L I Rudin, S Osher, and E Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.
- [39] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.
- [40] A. Buades, B. Coll, and J. M Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65 vol. 2, 2005.

APPENDIX A

RANDOM PATTERN GENERATING CODE

The code given below creates random patterns and saves them as .png files.

```
function [ ] = Matrix_Creator( )

SIZE=10; % Please enter the number of pixles on one side
%of matrix
TNM=100; % Please enter the total number of matrix you want
% to create
MULT=32; % Please enter the desired pixel size in milli-
%eters (DEPENDS ON HOW YOU PRODUCE THEM THOUGH)
for NOM=1:TNM
I=uint8(zeros(SIZE,SIZE));
j=0;
m=0;
M=rand(SIZE,SIZE);
MAT=zeros(SIZE*MULT);
N=M;
t1=sum(sum(M))/(SIZE*SIZE);
t2=t1;
t3=t1;
for i=1:SIZE*SIZE
N(i) = N(i)>t1;
end
K=N;
Sum=sum(sum(N));
Sum2=Sum;
Sum3=Sum;
Sum4=Sum;
if(Sum>SIZE*SIZE/2) %increase threshold
while(Sum2>SIZE*SIZE/2)
j=j+1;
t2=t1+j/(SIZE*1000);
for k=1:SIZE*SIZE
```

```

K(k) = M(k)>t2;
end
Sum2=sum(sum(K));
end
elseif(Sum<SIZE*SIZE/2) %decrease threshold
while(Sum3<SIZE*SIZE/2)
m=m+1;
t3=t1-m/(SIZE*1000);
for n=1:SIZE*SIZE
K(n) = M(n)>t3;
end
Sum3=sum(sum(K));
end
else
Sum4=Sum;
end
SumF=sum(sum(K));
if(SumF==SIZE*SIZE/2)
for idx1=1:SIZE*MULT
for idx2=1:SIZE*MULT
idx3=round((idx1-1)/MULT+0.5);
idx4=round((idx2-1)/MULT+0.5);
MAT((idx1-1)*SIZE*MULT+idx2)=K((idx3-1)*SIZE+idx4);
end
end
for idx1=1:SIZE*MULT
for idx2=1:SIZE*MULT
flag1=rem(idx1,MULT);
flag2=rem(idx1-1,MULT);
flag3=rem(idx2,MULT);
flag4=rem(idx2-1,MULT);
if (flag1 & flag2 & flag3 & flag4)
else
MAT((idx1-1)*SIZE*MULT+idx2)=1;
end
end
end
for o=1:SIZE*MULT
for p=1:SIZE*MULT
I(o,p)=MAT(SIZE*MULT*(o-1)+p)*255;
end
end
Add_1=ceil(size(I,1)/2); % OR CHANGE ADD TO ANYTHING YOU
%WANT
Add_2=ceil(size(I,2)/2); % OR CHANGE ADD TO ANYTHING YOU
%WANT
Framed_I=ones(2*Add_1+size(I,1),2*Add_2+size(I,2));

```



```
for i=1:size(I,1)
for j=1:size(I,2)
Framed_I(i+Add_1,j+Add_2)=I(i,j);
end
end
fname = strcat(num2str(NOM), '.png');
imwrite(Framed_I, fname, 'png')
end
end
end
```


APPENDIX B

IMAGE DISCRETIZATION CODE

The code given below discretizes a given image to defined resolution.

```
function [ J ] = Discretizor( I )
Size1=10;
Size2=10;
I2=double(I);
J=imresize(I2, [Size1 Size2], 'nearest');
end
```


APPENDIX C

CS IMAGE RECONSTRUCTION CODE

The code given below is the code used to reconstruct the ideal image. The main iterations given in line 231-235 are adopted from <http://users.ece.gatech.edu/~justin/l1magic/>

```
function [ ] = CALL_ALL( )
eps=1*10^(-3);
Data_Name_1='Ideal';           %Which data am I plotting?
Input1=[0.6074
0.6074
0.5477
0.4178
0.6348
0.4775
0.4845
0.3855
0.3616
0.5084
0.4178
0.5154
0.568
0.4522
0.4775
0.396
0.5751
0.5049
0.5716
0.66
0.4845
0.7232
0.6839
0.3363
0.6565
```

0.5477
0.481
0.4487
0.488
0.4178
0.2422
0.4283
0.3686
0.3855
0.5751
0.5751
0.4775
0.4283
0.4248
0.389
0.4522
0.6039
0.4775
0.5442
0.5645
0.6839
0.3855
0.3258
0.6635
0.66
0.396
0.5751
0.4452
0.5645
0.3925
0.6312
0.4178
0.5084
0.5154
0.6277
0.6909
0.7162
0.4248
0.4845
0.3651
0.4178
0.5716
0.3581
0.5049
0.5084
0.6277
0.3363

```

0.5968
0.5716
0.2767
0.2135
0.4845
0.5372
0.6671
0.5119
0.6909
0.4522
0.6565
0.5645
0.396
0.389
0.488
0.4775
0.6039
0.8033
0.5049
0.5716
0.6039
0.5477
0.4775
0.6039
0.6277
0.5084
];
Folder_Date=datestr(now,'yyyymmdd');
Folder_Name_1=strcat('C:\Works\Master\Compressive_Sensing\',
    Folder_Date, '\results\', Data_Name_1, '\');
Method_Name1='l1eq\';
Method_Name2='l1qc\';
Method_Name3='l1dantzig\';
Method_Name4='tvqc\';
Method_Name5='tvdantzig\';
Method_Name11='l1eq_Filtered\';
Method_Name22='l1qc_Filtered\';
Method_Name33='l1dantzig_Filtered\';
Method_Name44='tvqc_Filtered\';
Method_Name55='tvdantzig_Filtered\';
mkdir(Folder_Name_1, Method_Name1)
mkdir(Folder_Name_1, Method_Name2)
mkdir(Folder_Name_1, Method_Name3)
mkdir(Folder_Name_1, Method_Name4)
mkdir(Folder_Name_1, Method_Name5)
mkdir(Folder_Name_1, Method_Name11)
mkdir(Folder_Name_1, Method_Name22)

```

```

mkdir(Folder_Name_1, Method_Name33)
mkdir(Folder_Name_1, Method_Name44)
mkdir(Folder_Name_1, Method_Name55)
Matrices=uint8(zeros(320,320,100));
Size=max(size(Input1))+2;
Signal_Expectations=Input1;
for i=2:9
Gaa= imread([num2str(i)],'jpeg');
Matrices(:,:,i)=Gaa(:,:,1);
end
for i=11:Size
Gaa= imread([num2str(i)],'jpeg');
Matrices(:,:,i)=Gaa(:,:,1);
end
New_Matrix=false(10,10,Size);
for k=1:Size
for i=1:10
for j=1:10
New_Matrix(i,j,k)=1-Matrices((i-1)*32+16,(j-1)*32+16,k);
end
end
end
Size2=100;
for k=2:9
for i=1:10
for j=1:10
BP_Matrix(k-1, (i)+(j-1)*10)=New_Matrix(i,j,k);
end
end
end
ss1=zeros(1, max(size(Input1)));
ss2=zeros(1, max(size(Input1)));
ss3=zeros(1, max(size(Input1)));
ss4=zeros(1, max(size(Input1)));
ss5=zeros(1, max(size(Input1)));
ss11=zeros(1, max(size(Input1)));
ss22=zeros(1, max(size(Input1)));
ss33=zeros(1, max(size(Input1)));
ss44=zeros(1, max(size(Input1)));
ss55=zeros(1, max(size(Input1)));
mult1=zeros(1, max(size(Input1)));
mult2=zeros(1, max(size(Input1)));
mult3=zeros(1, max(size(Input1)));
mult4=zeros(1, max(size(Input1)));
mult5=zeros(1, max(size(Input1)));
mult11=zeros(1, max(size(Input1)));
mult22=zeros(1, max(size(Input1)));

```



```

mult33=zeros(1, max(size(Input1)));
mult44=zeros(1, max(size(Input1)));
mult55=zeros(1, max(size(Input1)));
only_R_1=zeros(1,max(size(Input1)));
only_R_2=zeros(1,max(size(Input1)));
only_R_3=zeros(1,max(size(Input1)));
only_R_4=zeros(1,max(size(Input1)));
only_R_5=zeros(1,max(size(Input1)));
only_R_11=zeros(1,max(size(Input1)));
only_R_22=zeros(1,max(size(Input1)));
only_R_33=zeros(1,max(size(Input1)));
only_R_44=zeros(1,max(size(Input1)));
only_R_55=zeros(1,max(size(Input1)));
for control_par=9:98
Size2=control_par+2;
for k=11:Size2
for i=1:10
for j=1:10
BP_Matrix(k-2, (i)+(j-1)*10)=New_Matrix(i,j,k);
end
end
end
n=100;
p=Size2-2;
A=double(BP_Matrix);
y=double(Signal_Expectations(1:p));
xp=[1:100]';
x1=l1eq_pd(xp,A,A,y,eps);
x2=l1qc_logbarrier(xp,A,A,y,eps);
x3=l1dantzig_pd(xp,A,A,y,eps);
x4=tvqc_logbarrier(xp,A,A,y,eps);
x5=tvdantzig_logbarrier(xp,A,A,y,eps);
Answer1=zeros(10,10);
for i=1:10
for j=1:10
Answer1(i,j)=x1(i+(j-1)*10);
end
end
Answer2=zeros(10,10);
for i=1:10
for j=1:10
Answer2(i,j)=x2(i+(j-1)*10);
end
end
Answer3=zeros(10,10);
for i=1:10
for j=1:10

```

```

Answer3(i,j)=x3(i+(j-1)*10);
end
end
Answer4=zeros(10,10);
for i=1:10
for j=1:10
Answer4(i,j)=x4(i+(j-1)*10);
end
end
Answer5=zeros(10,10);
for i=1:10
for j=1:10
Answer5(i,j)=x5(i+(j-1)*10);
end
end
MIN1=double(min(min(Answer1)));
MAX1=double(max(max(Answer1)));
MIN2=double(min(min(Answer2)));
MAX2=double(max(max(Answer2)));
MIN3=double(min(min(Answer3)));
MAX3=double(max(max(Answer3)));
MIN4=double(min(min(Answer4)));
MAX4=double(max(max(Answer4)));
MIN5=double(min(min(Answer5)));
MAX5=double(max(max(Answer5)));
Answer11=(double(Answer1)-MIN1)/(MAX1-MIN1);
Answer22=(double(Answer2)-MIN2)/(MAX2-MIN2);
Answer33=(double(Answer3)-MIN3)/(MAX3-MIN3);
Answer44=(double(Answer4)-MIN4)/(MAX4-MIN4);
Answer55=(double(Answer5)-MIN5)/(MAX5-MIN5);
MEAN11=sum(sum(Answer11))/size(Answer11,1)/size(Answer11,2);
MEAN22=sum(sum(Answer22))/size(Answer22,1)/size(Answer22,2);
MEAN33=sum(sum(Answer33))/size(Answer33,1)/size(Answer33,2);
MEAN44=sum(sum(Answer44))/size(Answer44,1)/size(Answer44,2);
MEAN55=sum(sum(Answer55))/size(Answer55,1)/size(Answer55,2);
Answer111=uint8(255*Answer11*0.5/MEAN11);
Answer222=uint8(255*Answer22*0.5/MEAN22);
Answer333=uint8(255*Answer33*0.5/MEAN33);
Answer444=uint8(255*Answer44*0.5/MEAN44);
Answer555=uint8(255*Answer55*0.5/MEAN55);
ZZZ1=imresize(Answer111,[320 320],'nearest');
ZZZ2=imresize(Answer222,[320 320],'nearest');
ZZZ3=imresize(Answer333,[320 320],'nearest');
ZZZ4=imresize(Answer444,[320 320],'nearest');
ZZZ5=imresize(Answer555,[320 320],'nearest');
YYY=imread('IDEAL_Image.png','PNG');
ss1(control_par)=SSIM(YYY,ZZZ1,[0.01 0.03],fspecial(

```

```

'gaussian', 250, 25));
ss2(control_par)=SSIM(YYY,ZZZ2, [0.01 0.03] , fspecial(
'gaussian', 250, 25));
ss3(control_par)=SSIM(YYY,ZZZ3, [0.01 0.03] , fspecial(
'gaussian', 250, 25));
ss4(control_par)=SSIM(YYY,ZZZ4, [0.01 0.03] , fspecial(
'gaussian', 250, 25));
ss5(control_par)=SSIM(YYY,ZZZ5, [0.01 0.03] , fspecial(
'gaussian', 250, 25));
mult1(control_par)=sum(sum((double(YYY)-double(ZZZ1)).^2))/
320/320/255/255;
mult2(control_par)=sum(sum((double(YYY)-double(ZZZ2)).^2))/
320/320/255/255;
mult3(control_par)=sum(sum((double(YYY)-double(ZZZ3)).^2))/
320/320/255/255;
mult4(control_par)=sum(sum((double(YYY)-double(ZZZ4)).^2))/
320/320/255/255;
mult5(control_par)=sum(sum((double(YYY)-double(ZZZ5)).^2))/
320/320/255/255;
only_R_1(control_par)=sum(sum(double(boolean(YYY)).*
(double(YYY)-double(ZZZ1)).^2))/255/255/
sum(sum(boolean(YYY))));
only_R_2(control_par)=sum(sum(double(boolean(YYY)).*
(double(YYY)-double(ZZZ2)).^2))/255/255/
sum(sum(boolean(YYY))));
only_R_3(control_par)=sum(sum(double(boolean(YYY)).*
(double(YYY)-double(ZZZ3)).^2))/255/255/
sum(sum(boolean(YYY))));
only_R_4(control_par)=sum(sum(double(boolean(YYY)).*
(double(YYY)-double(ZZZ4)).^2))/255/255/
sum(sum(boolean(YYY))));
only_R_5(control_par)=sum(sum(double(boolean(YYY)).*
(double(YYY)-double(ZZZ5)).^2))/255/255/
sum(sum(boolean(YYY))));
ZZZ11=ZZZ1;
ZZZ22=ZZZ2;
ZZZ33=ZZZ3;
ZZZ44=ZZZ4;
ZZZ55=ZZZ5;
for i=1:320*320
if ZZZ1(i)<128
ZZZ11(i)=0;
end
end
for i=1:320*320
if ZZZ2(i)<128
ZZZ22(i)=0;

```

```

end
end
for i=1:320*320
if ZZZ3(i)<128
ZZZ33(i)=0;
end
end
for i=1:320*320
if ZZZ4(i)<128
ZZZ44(i)=0;
end
end
for i=1:320*320
if ZZZ5(i)<128
ZZZ55(i)=0;
end
end
ss11(control_par)=SSIM(YYY,ZZZ11, [0.01 0.03] , fspecial(
'gaussian', 250, 25));
ss22(control_par)=SSIM(YYY,ZZZ22, [0.01 0.03] , fspecial(
'gaussian', 250, 25));
ss33(control_par)=SSIM(YYY,ZZZ33, [0.01 0.03] , fspecial(
'gaussian', 250, 25));
ss44(control_par)=SSIM(YYY,ZZZ44, [0.01 0.03] , fspecial(
'gaussian', 250, 25));
ss55(control_par)=SSIM(YYY,ZZZ55, [0.01 0.03] , fspecial(
'gaussian', 250, 25));
mult11(control_par)=sum(sum((double(YYY)-double(ZZZ11)).^2))/
320/320/255/255;
mult22(control_par)=sum(sum((double(YYY)-double(ZZZ22)).^2))/
320/320/255/255;
mult33(control_par)=sum(sum((double(YYY)-double(ZZZ33)).^2))/
320/320/255/255;
mult44(control_par)=sum(sum((double(YYY)-double(ZZZ44)).^2))/
320/320/255/255;
mult55(control_par)=sum(sum((double(YYY)-double(ZZZ55)).^2))/
320/320/255/255;
only_R_11(control_par)=sum(sum(double(boolean(YYY))
.*(double(YYY)-double(ZZZ11)).^2))/255/255/sum(
sum(boolean(YYY)));
only_R_22(control_par)=sum(sum(double(boolean(YYY))
.*(double(YYY)-double(ZZZ22)).^2))/255/255/sum(
sum(boolean(YYY)));
only_R_33(control_par)=sum(sum(double(boolean(YYY))
.*(double(YYY)-double(ZZZ33)).^2))/255/255/sum(
sum(boolean(YYY)));
only_R_44(control_par)=sum(sum(double(boolean(YYY))

```

```

.*(double(YYY)-double(ZZZ44)).^2))/255/255/sum(
sum(boolean(YYY)));
only_R_55(control_par)=sum(sum(double(boolean(YYY))
.*(double(YYY)-double(ZZZ55)).^2))/255/255/sum(
sum(boolean(YYY)));
fname1_1=strcat(Folder_Name_1, Method_Name1, Data_Name_1,
'_1_reconstructed_using_l1eq_pd_',
num2str(control_par), '.png');
fname1_2=strcat(Folder_Name_1, Method_Name2, Data_Name_1,
'_2_reconstructed_using_l1qc_logbarrier_',
num2str(control_par), '.png');
fname1_3=strcat(Folder_Name_1, Method_Name3, Data_Name_1,
'_3_reconstructed_using_l1dantzig_pd_',
num2str(control_par), '.png');
fname1_4=strcat(Folder_Name_1, Method_Name4, Data_Name_1,
'_4_reconstructed_using_tvqc_logbarrier_',
num2str(control_par), '.png');
fname1_5=strcat(Folder_Name_1, Method_Name5, Data_Name_1,
'_5_reconstructed_using_tvdantzig_logbarrier_',
num2str(control_par), '.png');
fname1_11=strcat(Folder_Name_1, Method_Name11, Data_Name_1,
'_11_filtered_reconstructed_using_l1eq_pd_',
num2str(control_par), '.png');
fname1_22=strcat(Folder_Name_1, Method_Name22, Data_Name_1,
'_22_filtered_reconstructed_using_l1qc_logbarrier_',
num2str(control_par), '.png');
fname1_33=strcat(Folder_Name_1, Method_Name33, Data_Name_1,
'_33_filtered_reconstructed_using_l1dantzig_pd_',
num2str(control_par), '.png');
fname1_44=strcat(Folder_Name_1, Method_Name44, Data_Name_1,
'_44_filtered_reconstructed_using_tvqc_logbarrier_',
num2str(control_par), '.png');
fname1_55=strcat(Folder_Name_1, Method_Name55, Data_Name_1,
'_55_filtered_reconstructed_using_tvdantzig_logbarrier_',
num2str(control_par), '.png');
imwrite(ZZZ1, fname1_1, 'png')
imwrite(ZZZ2, fname1_2, 'png')
imwrite(ZZZ3, fname1_3, 'png')
imwrite(ZZZ4, fname1_4, 'png')
imwrite(ZZZ5, fname1_5, 'png')
imwrite(ZZZ11, fname1_11, 'png')
imwrite(ZZZ22, fname1_22, 'png')
imwrite(ZZZ33, fname1_33, 'png')
imwrite(ZZZ44, fname1_44, 'png')
imwrite(ZZZ55, fname1_55, 'png')
end
pname1_1=strcat(Folder_Name_1, 'progress_of_' , Data_Name_1,

```

```

'_1_reconstructed_using_l1eq_pd_', '.png');
pname1_2=strcat(Folder_Name_1, 'progress_of_' , Data_Name_1,
'_2_reconstructed_using_l1qc_logbarrier', '.png');
pname1_3=strcat(Folder_Name_1, 'progress_of_' , Data_Name_1,
'_3_reconstructed_using_l1dantzig_pd', '.png');
pname1_4=strcat(Folder_Name_1, 'progress_of_' , Data_Name_1,
'_4_reconstructed_using_tvqc_logbarrier', '.png');
pname1_5=strcat(Folder_Name_1, 'progress_of_' , Data_Name_1,
'_5_reconstructed_using_tvdantzig_logbarrier', '.png');
pname1_11=strcat(Folder_Name_1, 'progress_of_' , Data_Name_1,
'_11_filtered_reconstructed_using_l1eq_pd_', '.png');
pname1_22=strcat(Folder_Name_1, 'progress_of_' , Data_Name_1,
'_22_filtered_reconstructed_using_l1qc_logbarrier', '.png');
pname1_33=strcat(Folder_Name_1, 'progress_of_' , Data_Name_1,
'_33_filtered_reconstructed_using_l1dantzig_pd', '.png');
pname1_44=strcat(Folder_Name_1, 'progress_of_' , Data_Name_1,
'_44_filtered_reconstructed_using_tvqc_logbarrier', '.png');
pname1_55=strcat(Folder_Name_1, 'progress_of_' , Data_Name_1,
'_55_filtered_reconstructed_using_tvdantzig_logbarrier', '.png');
rname1_1=strcat(Folder_Name_1,
'only_R_progress_of_' , Data_Name_1,
'_1_reconstructed_using_l1eq_pd_', '.png');
rname1_2=strcat(Folder_Name_1,
'only_R_progress_of_' , Data_Name_1,
'_2_reconstructed_using_l1qc_logbarrier', '.png');
rname1_3=strcat(Folder_Name_1,
'only_R_progress_of_' , Data_Name_1,
'_3_reconstructed_using_l1dantzig_pd', '.png');
rname1_4=strcat(Folder_Name_1,
'only_R_progress_of_' , Data_Name_1,
'_4_reconstructed_using_tvqc_logbarrier', '.png');
rname1_5=strcat(Folder_Name_1,
'only_R_progress_of_' , Data_Name_1,
'_5_reconstructed_using_tvdantzig_logbarrier', '.png');
rname1_11=strcat(Folder_Name_1,
'only_R_progress_of_' , Data_Name_1,
'_11_filtered_reconstructed_using_l1eq_pd_', '.png');
rname1_22=strcat(Folder_Name_1,
'only_R_progress_of_' , Data_Name_1,
'_22_filtered_reconstructed_using_l1qc_logbarrier', '.png');
rname1_33=strcat(Folder_Name_1,
'only_R_progress_of_' , Data_Name_1,
'_33_filtered_reconstructed_using_l1dantzig_pd', '.png');
rname1_44=strcat(Folder_Name_1,
'only_R_progress_of_' , Data_Name_1,
'_44_filtered_reconstructed_using_tvqc_logbarrier', '.png');
rname1_55=strcat(Folder_Name_1,

```

```

'only_R_progress_of_' , Data_Name_1,
'_55_filtered_reconstructed_using_tvdantzig_logbarrier',
'.png');
sname1_1=strcat(Folder_Name_1, 'ss_progress_of_' , Data_Name_1,
'_1_reconstructed_using_l1eq_pd_' , '.png');
sname1_2=strcat(Folder_Name_1, 'ss_progress_of_' , Data_Name_1,
'_2_reconstructed_using_l1qc_logbarrier' , '.png');
sname1_3=strcat(Folder_Name_1, 'ss_progress_of_' , Data_Name_1,
'_3_reconstructed_using_l1dantzig_pd' , '.png');
sname1_4=strcat(Folder_Name_1, 'ss_progress_of_' , Data_Name_1,
'_4_reconstructed_using_tvqc_logbarrier' , '.png');
sname1_5=strcat(Folder_Name_1, 'ss_progress_of_' , Data_Name_1,
'_5_reconstructed_using_tvdantzig_logbarrier' , '.png');
sname1_11=strcat(Folder_Name_1, 'ss_progress_of_' , Data_Name_1,
'_11_filtered_reconstructed_using_l1eq_pd_' , '.png');
sname1_22=strcat(Folder_Name_1, 'ss_progress_of_' , Data_Name_1,
'_22_filtered_reconstructed_using_l1qc_logbarrier' , '.png');
sname1_33=strcat(Folder_Name_1, 'ss_progress_of_' , Data_Name_1,
'_33_filtered_reconstructed_using_l1dantzig_pd' , '.png');
sname1_44=strcat(Folder_Name_1, 'ss_progress_of_' , Data_Name_1,
'_44_filtered_reconstructed_using_tvqc_logbarrier' , '.png');
sname1_55=strcat(Folder_Name_1, 'ss_progress_of_' , Data_Name_1,
'_55_filtered_reconstructed_using_tvdantzig_logbarrier' , '.png');
mult_mat=strcat(Folder_Name_1, 'progress_of_' , Data_Name_1,
'_all_progresses_of_mult' , '.mat');
only_R_mat=strcat(Folder_Name_1, 'progress_of_' , Data_Name_1,
'_all_progresses_of_only_R' , '.mat');
ss_mat=strcat(Folder_Name_1, 'progress_of_' , Data_Name_1,
'_all_progresses_of_ss' , '.mat');
save(mult_mat, 'mult1', 'mult2', 'mult3', 'mult4',
'mult5', 'mult11', 'mult22', 'mult33', 'mult44', 'mult55')
save(only_R_mat, 'only_R_1', 'only_R_2', 'only_R_3', 'only_R_4',
'only_R_5', 'only_R_11', 'only_R_22', 'only_R_33', 'only_R_44',
'only_R_55')
save(ss_mat, 'ss1', 'ss2', 'ss3', 'ss4', 'ss5',
'ss11', 'ss22', 'ss33', 'ss44', 'ss55')
plot(mult1)
print('-dpng', pname1_1)
plot(mult2)
print('-dpng', pname1_2)
plot(mult3)
print('-dpng', pname1_3)
plot(mult4)
print('-dpng', pname1_4)
plot(mult5)
print('-dpng', pname1_5)
plot(mult11)

```

```

print('-dpng', pname1_11)
plot(mult22)
print('-dpng', pname1_22)
plot(mult33)
print('-dpng', pname1_33)
plot(mult44)
print('-dpng', pname1_44)
plot(mult55)
print('-dpng', pname1_55)
plot(only_R_1)
print('-dpng', rname1_1)
plot(only_R_2)
print('-dpng', rname1_2)
plot(only_R_3)
print('-dpng', rname1_3)
plot(only_R_4)
print('-dpng', rname1_4)
plot(only_R_5)
print('-dpng', rname1_5)
plot(only_R_11)
print('-dpng', rname1_11)
plot(only_R_22)
print('-dpng', rname1_22)
plot(only_R_33)
print('-dpng', rname1_33)
plot(only_R_44)
print('-dpng', rname1_44)
plot(only_R_55)
print('-dpng', rname1_55)
plot(ss1)
print('-dpng', sname1_1)
plot(ss2)
print('-dpng', sname1_2)
plot(ss3)
print('-dpng', sname1_3)
plot(ss4)
print('-dpng', sname1_4)
plot(ss5)
print('-dpng', sname1_5)
plot(ss11)
print('-dpng', sname1_11)
plot(ss22)
print('-dpng', sname1_22)
plot(ss33)
print('-dpng', sname1_33)
plot(ss44)
print('-dpng', sname1_44)

```



```
plot(ss55)
print('-dpng', sname1_55)
end
```


APPENDIX D

LINE SHIFTING CODE

The code given below is the code used to note the shifts in the line. These shifts are used in reconstruction in the code given in Appendix E.

```
function [ ] = Line_Shifts( )
Raw_Data = dir('*EMITFILE.CSV');
for i=1:length(Raw_Data)
eval(['load ' Raw_Data(i).name ' -ascii']);
end
BG_IA_Try=zeros(80,180);
for i=3:12
BG_IA_Try=BG_IA_Try+(eval(sprintf('X%dEMITFILE',i-1)))/10;
end
Tri_90_3_1_Raw=zeros(80,180);
for i=30:length(Raw_Data)
Tri_90_3_1_Raw=Tri_90_3_1_Raw+(eval(sprintf('X%dEMITFILE',i-1)))/
(length(Raw_Data)-30);
end
Diff1_IA=(Tri_90_3_1_Raw-BG_IA_Try);
MIN=min(min(Diff1_IA));
MAX=max(max(Diff1_IA));
Diff_IA=256*256-uint16((Tri_90_3_1_Raw-BG_IA_Try-MIN)*256*256/
(MAX-MIN));
imwrite(Diff_IA,'Raw_Data.png','PNG')
Rec=uint16(zeros(180,80));
for i=1:80
for j=1:180
Rec(j,i)=Diff_IA(i,j);
end
end
Tri_90_3_1_Proc=Rec;
AIM_SIZE1=180;
AIM_SIZE2=80;
Data_resized=imresize(Tri_90_3_1_Proc, [AIM_SIZE1 AIM_SIZE2]);
```

```

DATA_RAW=Data_resized;
for i=1:AIM_SIZE1
for j=1:AIM_SIZE2
if (Data_resized(i,j) < 0.3*256*256)
DATA_RAW(i,j)=0;
end
end
end
imwrite(DATA_RAW, 'DATA_RAW.png', 'PNG')
INPUT_IM=uint16(imread('Bez_tytulu.png'));
OUTPUT_IM=uint16(zeros(AIM_SIZE1,AIM_SIZE2));
PROC_IM=INPUT_IM(62:1097,237:710);
OUTPUT_IM=imresize(PROC_IM, [AIM_SIZE1 AIM_SIZE2]);
Aim_on_screen=uint16(OUTPUT_IM);
imwrite(Aim_on_screen,'Aim_on_screen.png','PNG')
AIM_RAW=Aim_on_screen*256;
imwrite(AIM_RAW, 'AIM_RAW.png', 'PNG')
Diff_Init=0;
for m=1:AIM_SIZE1
for n=1:AIM_SIZE2
Diff_Init=Diff_Init+double(abs(DATA_RAW(m,n)-AIM_RAW(m,n)));
end
end
DATA_RAW_SHIFTED=uint16(zeros(AIM_SIZE1,AIM_SIZE2));
Shifted_Data=uint16(zeros(AIM_SIZE1,AIM_SIZE2));
Shifted_Data=DATA_RAW;
Temp=uint16(zeros(AIM_SIZE1,AIM_SIZE2));
Diff_SD=double(zeros(AIM_SIZE2,AIM_SIZE2));
for i=1:AIM_SIZE2
for j=1:AIM_SIZE2
Shifted_Data=DATA_RAW;
Shifted_Data(:,i)=Shifted_Data(:,j);
Diff_SD(i,j)=double(sum(sum(abs(Shifted_Data-AIM_RAW))));
end
end
Delete_this=zeros(AIM_SIZE2,1);
Replace_this=zeros(AIM_SIZE2,1);
for j=1:AIM_SIZE2
[r,c]=find(Diff_SD==min(min(Diff_SD)));
change_var1=r(1); %DESTROY THIS
change_var2=c(1); %PLACE THIS IN ITS PLACE
if(Diff_SD(r(1),c(1))<Diff_Init)
DATA_RAW_SHIFTED(:,change_var1)=DATA_RAW(:,change_var2);
Delete_this(j,1)=r(1);
Replace_this(j,1)=c(1);
end
Diff_SD(change_var1,:)=271*124*256*256*3;

```

```
Diff_SD(:,change_var2)=271*124*256*256*3;
end
xlswrite('Delete_this',Delete_this)
xlswrite('Replace_this',Replace_this)
imwrite(DATA_RAW_SHIFTED, 'DATA_RAW_SHIFTED.png', 'PNG');
end
```


APPENDIX E

PASSIVE IMAGING SYSTEM IMAGE RECONSTRUCTION CODE

The code given below is the code used to reconstruct the image from the raw data given from the passive imaging system. Unfortunately the raw data cannot be given here, interested readers should contact author for further information.

```
function [ I_double ] = EMITFILE2RAW_IMAGE(h_for_NO_White_Lines)
% INPUT SOME AVERAGING FUNCTION like
%h=fspecial('gaussian', [1 5], 1);
% This code is developed to convert more than 40 and less than 99
% CSV files taken from the
% Passive imaging System's "TVF_" option.
% This code takes some of the first
%CSV files as a background and some of
% the last CSV files as the data.
%It subtracts background from data to
% show the image number 3.
%BUT as you can see this image is meaningless
% since it is not a normalized image,
%ie. the max is not 1 and the min is
% not 0. Therefore the code normalizes the image to
%this range (to 0-65536
% actually, step 4)
% Then it takes a 135 degrees mirror image because
% this is the true image
% (step5).
% Then it RE-shuffles the lines because the raw data is
%shuffles in
% some random order we could not understand (step 6)
% Then finally it averages some extremely bright lines,
% again, we could not
% understand why they are so bright (step 7)
```

```

% UPDATE now it also saves the double version of image,
% and returns that
%
% -----0-----
%
% NOTES:
% 1) You must run this code where there are MORE THAN 40 CSV FILES!
% This is a must since the first data are just background so
%no image can
% be formed with these...
% 2) you must run this code where there are
%LESS THAN 99 CSV FILES!
% This is a must since if you take data for too long
%it overwrites the
% first datas so you lose background. If there is no
%background then code
% cannot substract the background to form image.
%THEREFORE IT IS VERY
% IMPORTANT TO TAKE DATA FOR ONLY A FEW SECONDS.
%IT IS HIGHLY RECOMMENDED
% THAT YOU CHECK THE "C:\TVNetwork" DIRECTORY WHEN YOU ARE
% TAKING DATA TO
% MAKE SURE YOU STOP TAKING DATA BEFORE YOU REACH 99EMITFILE.CSV
test1=dir('*EMITFILE.CSV');
test2=size(test1);
if (test2(1)<40)
    fprintf ( 1, '\n');
    fprintf ( 1, 'You need to call this function in a directory
        where there are MORE than 40 *EMITFILES\n');
    return;
endtest3=dir('*EMITFILE.CSV');
test4=size(test3);
if (test4(1)>98)
    fprintf ( 1, '\n');
    fprintf ( 1, 'You need to call this function in a directory
        where there are LESS than 99 *EMITFILES\n');
    return;
end
replacer=[9
1
10
2
11
3
4
12
17

```


13
5
73
18
14
6
74
15
7
19
75
20
16
8
21
25
76
65
26
22
77
66
23
27
78
67
79
28
24
68
80
29
33
69
57
30
34
70
58
31
35
71
59
32
36
72
60

```

41
49
37
61
42
38
50
43
62
39
44
63
51
40
52
64
45
53
46
54
47
55
48
56];
Raw_Data = dir('*EMITFILE.CSV');
for i=1:length(Raw_Data)
eval(['load ' Raw_Data(i).name ' -ascii']);
end
Data_BG_av=double(zeros(80,180));
for i=8:16
Data_BG_av=Data_BG_av+double((eval(sprintf
('X%dEMITFILE',i-1))))/9;
end
imwrite(Data_BG_av/256/256, '_1Data_BackGround_avaraged', 'PNG')
Data_Raw_av=double(zeros(80,180));
for i=40:length(Raw_Data)
Data_Raw_av=Data_Raw_av+double((eval(sprintf('X%dEMITFILE',i-1)))/
(length(Raw_Data)-40));
end
imwrite(Data_Raw_av/256/256, '_2Data_image_avaraged', 'PNG')
IMAGE_RAW=double(zeros(80,180));
IMAGE_RAW=double((Data_Raw_av-Data_BG_av));
imwrite((IMAGE_RAW+256*256)/256/256/2,
'_3image-BackGround', 'PNG')
MIN=double(min(min(IMAGE_RAW)));
MAX=double(max(max(IMAGE_RAW)));
IMAGE_RAW_normalized=uint16(zeros(80,180));

```

```

IMAGE_RAW_normalized=uint16(256*256-(Data_Raw_av-Data_BG_av-
MIN)*256*256/(MAX-MIN));
imwrite(IMAGE_RAW_normalized,
'_4image-BackGround_(Normalized)', 'PNG')
IMAGE=uint16(zeros(180,80));
for i=1:80
for j=1:180
IMAGE(j,i)=IMAGE_RAW_normalized(i,j);
end
end
imwrite(IMAGE, '_5IMAGE_shuffled', 'PNG')
DATA_RAW_SHIFTED=uint16(zeros(180, 80));
for j=1:80
change_var1=j; %DESTROY THIS
change_var2=replacer(j); %PLACE THIS IN ITS PLACE
if(change_var1)
if(change_var2)
DATA_RAW_SHIFTED(:,change_var1)=IMAGE(:,change_var2);
end
end
end
imwrite(DATA_RAW_SHIFTED, '_6IMAGE', 'PNG');
BG_Normalized=double(DATA_RAW_SHIFTED);
Sizes=size(BG_Normalized);
Size1=Sizes(1);
Size2=Sizes(2);
Brightness=double(zeros(Size1,Size2));
for i=1:Size1
for j=1:Size2
Brightness(i,j)=(DATA_RAW_SHIFTED(Size1,j)
+DATA_RAW_SHIFTED(1,j))/2;
end
end
for i=1:Size2
BG_Normalized(:,i)=double(DATA_RAW_SHIFTED(:,i))-
double(Brightness(:,i));
end
MIN=double(min(min(BG_Normalized)));
MAX=double(max(max(BG_Normalized)));
BG_Normalized_normalized=uint16(zeros(80,180));
BG_Normalized_normalized=uint16(
(double(BG_Normalized)-MIN)*256*256/
(MAX-MIN));
imwrite(BG_Normalized_normalized, '_7IMAGE_background_normalized',
'PNG');
h_wo_White_Lines=h_for_NO_White_Lines;
NO_White_Lines2=conv2(double(BG_Normalized_normalized),

```

```

h_wo_White_Lines, 'same');
DATA_wo_White_Lines2=BG_Normalized_normalized;
DATA_wo_White_Lines2(:,6)=NO_White_Lines2(:,6);
DATA_wo_White_Lines2(:,19)=NO_White_Lines2(:,19);
DATA_wo_White_Lines2(:,33)=NO_White_Lines2(:,33);
DATA_wo_White_Lines2(:,35)=NO_White_Lines2(:,35);
DATA_wo_White_Lines2(:,50)=NO_White_Lines2(:,50);
DATA_wo_White_Lines2(:,66)=NO_White_Lines2(:,66);
DATA_wo_White_Lines2(:,69)=NO_White_Lines2(:,69);
MIN=double(min(min(DATA_wo_White_Lines2)));
MAX=double(max(max(DATA_wo_White_Lines2)));
DATA_wo_White_Lines_normalized2=uint16(zeros(80,180));
DATA_wo_White_Lines_normalized2=uint16(
(double(DATA_wo_White_Lines2)-MIN)*256*256/(MAX-MIN));
imwrite(DATA_wo_White_Lines_normalized2,
'_8IMAGE_wo_White_Lines', 'PNG');
I_double=zeros(180,80);
I_double=double(DATA_wo_White_Lines_normalized2)/256/256;
imwrite(I_double, '_9IMAGE_wo_White_Lines_double', 'PNG');
end

```

APPENDIX F

IMAGE AVERAGING CODE

The code given below is the code used to average the raw image. The averaging window is an input of the code. In this work the input was chosen to be $h1 = fspecial('gaussian', [15], 2)$.

```
function [Averaged_Combined_normalized] =  
Averaging(I, h_Averaging)  
h=h_Averaging;  
Averaged_Image=conv2(I,h,'same');  
MIN=double(min(min(Averaged_Image)));  
MAX=double(max(max(Averaged_Image)));  
Averaged_Combined_normalized=double(zeros(size(I,1),size(I,2)));  
Averaged_Combined_normalized=double((double(Averaged_Image)-MIN)/  
(MAX-MIN));  
imwrite(Averaged_Combined_normalized,  
 '_13Averaged_Image_normalized.png', 'PNG');  
end
```


APPENDIX G

ACTIVE IMAGING SIMULATION ACCORDING TO PSF

The code given below simulates the active imaging system according to the PSFs provided at 9 points. Unfortunately the PSFs cannot be given here, interested readers should contact author for further information.

```
function [ J ] = PSF( )
PSF(:, :, 1) = xlsread('m8m9.xlsx');
PSF(:, :, 2) = xlsread('0m9.xlsx');
PSF(:, :, 3) = xlsread('8m9.xlsx');
PSF(:, :, 4) = xlsread('m80.xlsx');
PSF(:, :, 5) = xlsread('00.xlsx');
PSF(:, :, 6) = xlsread('80.xlsx');
PSF(:, :, 7) = xlsread('m89.xlsx');
PSF(:, :, 8) = xlsread('09.xlsx');
PSF(:, :, 9) = xlsread('89.xlsx');
%I=double(imread('Figure_Act5.png'));
%for i=1:12
%   for j=1:12
%       I(i,j)=(-1)^(i+j);
%   end
%end
f1=50;
f2=50;
p1=30;
p2=30;
Im_resized=imresize(I(:, :, 1), [floor(200*p1/6+5*200/6)
    floor(200*p2/6+5*200/6)], 'nearest');
%imshow(Im_resized)
Target=Im_resized(100:1066,100:1066);
Im_resized2=(Target-ones(size(Target,1),size(Target,
2))*min(min(Target)))/(max(max(Target))-min(min(Target)));
imwrite(Im_resized2, 'Target.png', 'png')
t1=f1/p1;
```

```

t2=f2/p2;
k=1;
for i=1:t1:f1
    l=1;
    for j=1:t2:f2
        var=[i j];
        Im_cur=Im_resized((floor((k-1)*200/6)+1):
            (floor((k-1)*200/6)+200),(floor((l-1)*200/6)+1):
            (floor((l-1)*200/6)+200));
x1=(var(1)-1)^2;
x2=(var(1)-1)^2;
x3=(var(1)-1)^2;
x4=(var(1)-25.5)^2;
x5=(var(1)-25.5)^2;
x6=(var(1)-25.5)^2;
x7=(var(1)-50)^2;
x8=(var(1)-50)^2;
x9=(var(1)-50)^2;
y1=(var(2)-1)^2;
y2=(var(2)-25.5)^2;
y3=(var(2)-50)^2;
y4=(var(2)-1)^2;
y5=(var(2)-25.5)^2;
y6=(var(2)-50)^2;
y7=(var(2)-1)^2;
y8=(var(2)-25.5)^2;
y9=(var(2)-50)^2;
L1=x1+y1;
L2=x2+y2;
L3=x3+y3;
L4=x4+y4;
L5=x5+y5;
L6=x6+y6;
L7=x7+y7;
L8=x8+y8;
L9=x9+y9;
Ls=transpose([L1 L2 L3 L4 L5 L6 L7 L8 L9]);
[B, IX] = sort(Ls);
PSF1=PSF(:, :, IX(1));
PSF2=PSF(:, :, IX(2));
PSF3=PSF(:, :, IX(3));
PSF_cur=(PSF1*(B(2)+0.01)*(B(3)+0.01)+PSF2*(B(1)+0.01)*(B(3)+0.01)
+PSF3*(B(2)+0.01)*(B(1)+0.01))/
((B(1)+0.01)*(B(2)+0.01)*(B(3)+0.01));
Av=sum(sum(PSF_cur));
J(k,1)=sum(sum(PSF_cur.*Im_cur))/Av;
l=l+1;

```



```
end
k=k+1;
end
J2=(J-min(min(J)))/(max(max(J))-min(min(J)));
imshow(J2)
imwrite(J2, 'Image_obt.png', 'png')
end
```


APPENDIX H

ACTIVE IMAGING SYSTEM SIMULATION WITH WRONG RECEIVER INTEGRATION TIME

The code given below simulates an imaging system with the receiver's integration time corresponding to 3 pixels.

```
function [ J2 ] = Systematic( Input )
%SYSTEMATIC Summary of this function goes here
% Detailed explanation goes here
N=30;
J3=ones(30,30);
I3=ones(30,30);
for i=1:30
    for j=1:30
        for k=1:30
            for l=1:30
                if (j>l)
                    J3(j,i)=J3(j,i)+I3(l,k)*
                    exp(-((i-k+1)*(j-l+1)/30+(j-l+1))/3)/N;
                elseif (j==l && i>k)
                    J3(j,i)=J3(j,i)+I3(l,k)*
                    exp(-((i-k+1)*(j-l+1)/30+(j-l+1))/3)/N;
                end
            end
        end
    end
end
I=double(Input);
J=I;
flag=0;

for i=1:30
    for j=1:30
        for k=1:30
```

```

    for l=1:30
        if (j>1)
            J(j,i)=J(j,i)+I(1,k)*
                exp(-((i-k+1)*(j-l+1)/30+(j-l+1))/3)/N;
            elseif (j==1 && i>k)
                J(j,i)=J(j,i)+I(1,k)*
                    exp(-((i-k+1)*(j-l+1)/30+(j-l+1))/3)/N;
            end
        end
    end
end
end
end
end
end
%flag
J4=J./J3;
J2=(J4-min(min(J4)))/(max(max(J4))-min(min(J4)));
imwrite(J2, 'noisy_sys.png', 'png')
end

```