

國立交通大學
光電工程研究所
碩士論文

THz 顯像技術之研究



研究生：巫嗣文

指導教授：潘犀靈 教授

中華民國九十二年七月

THz 顯像技術之研究

A Study of THz Imaging

研究生：巫嗣文

Student: Shih-Wen Wu

指導老師：潘犀靈

Advisor: Prof. Ci-Ling Pan

國立交通大學

光電工程研究所

碩士論文

A Dissertation

Submitted to Institute of Electro-Optical Engineering
College of Electrical Engineering and Computer Science

National Chiao Tung University

In Partial Fulfillment of the Requirements

For the Degree of

Master of Engineering

In

Electro-Optical Engineering

July 2003

Hsinchu, Taiwan, Republic of China

中華民國九十二年七月

中文摘要

在本論文中，我們架設了一個含有快速掃描的 THz Imaging 系統並介紹 THz 成像方式以及快速掃描，成像方式有固定時間延遲、不同時間延遲成像、頻域振幅成像、以及頻域相位成像，我們並且掃描了新台幣之浮水印及火柴盒。其快速掃描系統能達到以 33 秒鐘得到訊噪比 131，兩分鐘半達到 212，十四分鐘達到 436，最快掃描速度為以鎖相放大器放大器時間常數 1ms 掃描頻率為 3Hz。我們也初步對系統的解析度做分析，量取焦點之 THz 焦點大小，並改用小 gap 之天線來做 Imaging，其解析度約可達到 400 μm 。

Abstract

We constructed a THz Imaging system with fast scan, and introduce THz imaging and fastscan. Our means of Imaging include fixed time delay, different time delay, amplitude at different frequency, and phase at different frequency. We also scanned currency mark on New Taiwan Dollars and a match box. It can achieved a SNR of 131 in 33 seconds, 212 in two and half minutes, 436 in 14 minutes. The fastest scan rate is 3Hz with 1ms of LIA time constant. We also simply discuss the resolution of our system, measured the THz spot size at focal point, and used a small gap antenna to do imaging to improve its spatial resolution. Its resolution may achieve about 400 μm now.



致謝

首先感謝潘犀靈老師兩年來的教導，以及蔡宗儒、劉子安學長從旁協助。感謝李兆達、王怡超以及陳晉璋學長在使用雷射上的幫助，還有陳昭遠學長及其哥哥在使用數位擷取卡上的示範。另外也感謝郭威宏、林世宣、石宗盛、林建宏同學兩年中的陪伴與分享實驗經驗。還有王之揚、張學智、黃龍進、江秉其、陳霖霖、賴奕帆學弟學妹們處理實驗室的雜務。兩年的實驗室生活使我成長不少，謝謝大家陪我度過這兩年。



目錄：

中文摘要	I
英文摘要	II
致謝	III
第一章 緒論	
1.1 THz 之簡介	1
1.2 THz 成像之特性	2
1.3 THz 成像之應用	3
第二章 THz 的產生與偵測	
2.1 THz 的產生	5
光電導模型	5
光整流模型	9
2.2 THz 的偵測	13
電光取樣	15
光導天線偵測	21
2.3 系統中所採用之 THz 的產生與偵測	22
第三章 THz Imaging 與系統架構	

3.1 系統架構	25
3.2 固定時間延遲，峰值成像	30
3.3 不同時間延遲，電場值成像	32
3.4 頻域振幅作圖	34
3.5 頻域相位作圖	35

第四章 快速掃描

4.1 THz 成像之缺陷及改進	38
4.2 快速掃描	38
4.3 快速掃描之量取	42
4.4 Rescale	46
4.5 鎖相放大器時間常數 (time constant)的選擇	48
4.6 平均次數與訊噪比	51
4.7 以快速掃描作 THz 成像	55
4.8 像素 waveform 之討論	61



第五章 THz 成像解析度

5.1 THz 成像解析度的限制	66
5.2 本系統之解析度測試	66
5.3 改以小 gap 天線來作成像	70

第六章 結論與未來展望	73
附錄 A	74
附錄 B	90
參考文獻	107



圖表目錄

圖 1-1 THz 輻射在頻譜上的分佈	1
圖 2-1 THz emitter 示意圖	6
圖 2-2 以光整流效應產生 THz 及傅式轉換	12
圖 2-3 掃到的 THz waveform	13
圖 2-4 系統架構示意簡圖	14
圖 2-5 電光取樣架構示意圖	15
圖 2-6 $\cos^2 \frac{\varphi}{2}$ 、 $\sin^2 \frac{\varphi}{2}$ 與相位差的關係圖	18
圖 2-7 以 balanced detector 來取代偏極片之電光取樣架構圖	20
圖 2-8 本系統所量到之 THz waveform 及其傅式轉換	24
圖 3-1 THz Imaging 系統架構圖	26
圖 3-2 THz Imaging 系統照片	27
圖 3-3 光學延遲線照片及推動裝置示意圖	29
圖 3-4 以 THz Imaging 掃新台幣 100 元與 1000 元之浮水印	31
圖 4-1.a 以步進馬達來掃 THz 的 waveform	40
圖 4-1.bcd 以快速掃描的方式來掃 THz 的 waveform , 掃描時間為一分鐘、 兩分鐘 三分鐘	41
圖 4-2 快速掃描擷取信號的一部份與 500 筆平均	45
圖 4-3.a 步進馬達位置 -10.8mm 所掃 waveform	47

圖 4-3.b	Rescale 過後的 fastscan 與 slowscanned waveform 之比較	47
圖 4-3.c	Rescale 後傅氏轉換	48
圖 4-4	鎖相放大器不同時間常數下所取得的 waveform	49
圖 4-5	以不同時間常數下,不同掃描頻率所掃的 waveform	50
圖 4-6.abcde	不同次數 waveform 之平均	52
圖 4-6.f	訊噪比與平均筆數關係以及與 $N^{1/2}$ 之比較	55
圖 4-7.a	火柴盒內各處的 THz waveform	62
圖 4-7.b	空白-火柴棒處的 THz waveform 與空白及火柴棒相加之比較	63
圖 4-8	鋸齒狀的頻譜分佈	64
圖 4-9	$a'(\omega)$ 對不同 $\Delta\theta$ 的關係	65
圖 5-1	樣品實際照片	66
圖 5-2	以每點間距 200mm 來掃 LEO 三個字	67
圖 5-3	以每點間距 100mm 來掃 LEO 三個字	68
圖 5-4	以刀口法量取 spotsize	69
圖 5-5	以 5mm gap 所量到的 THz 信號與 10mm gap 的信號之比較	70
圖 5-6	以 5mm gap 所量到之 THz Imaging	71
圖 5-7	以刀口法量取小 gap 天線所產生 spotsize	72



第一章 緒論

1.1 THz 之簡介

THz 波(or far-infrared)是在電磁頻譜上介在紅外頻段與微波中間的一個頻段,其頻率分布範圍約從 0.1 至 10THz。從波長的角度來看,THz 波又被稱為次毫米波(sub-millimeter wave),這是由於其波長約略比毫米短些(1 THz 可換算為波長約 0.33mm) [1]。如下圖所示(取自 Zhang XC 網站 <http://www.rpi.edu/~zhangxc>)可看到 THz 在頻譜上的分佈。

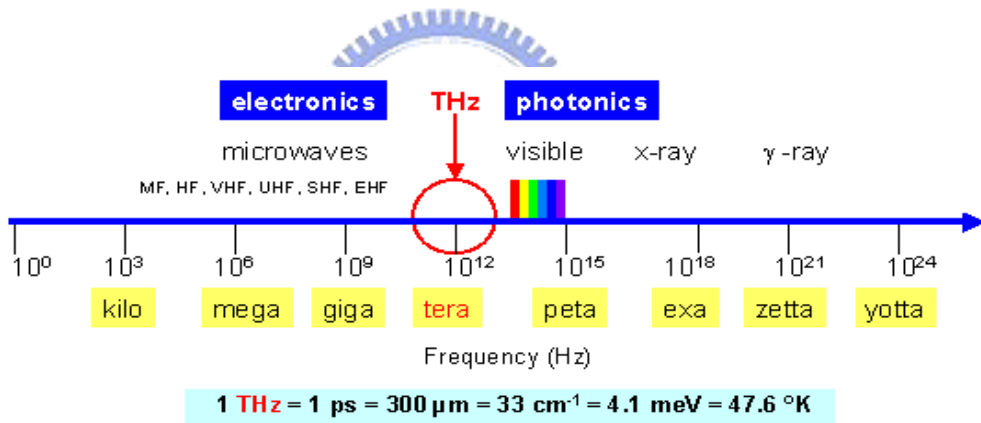


圖 1-1 THz 輻射在頻譜上的範圍

在凝態物理中,THz 波段是一個非常重要的頻譜。因為在 THz 這波段中,包含了許多決定材料特性的重要能階,例如半導體中受體(acceptor)、施體(donor)、以及光激子(exciton)等束縛能,光模聲子(optical phonon)、超導能隙,磁場作用下 Landau 能譜等,也都落在這

一波段中。又由於 THz 波段包含了大多數分子的轉動或振動能階，所以在無線電天文、遙測、醫學影像上之用途，也是非常有展望的。而隨著高速資訊時代之來臨，元件操作頻率的增加，也使得元件在 THz 波段的研究越來越重要，雖然 THz 波段具有如此重要的科學價值，但在此波段的研究，卻未如預期般的熱切展開。這最主要的問題就是，在 THz 波波段欠缺良好且經濟的波源及其相關元件[4]。

最近 THz 在各個領域上已被廣泛地研究，如 M. C. Nuss 等人[5]證明可以將 THz 應用在包裹檢測及生物體的化學成分析上。Buttner R 等人[6]利用 THz-TDS 頻譜技術觀察發現 DNA 在異種狀態 (hybridized/denatured) 下其的複數折射率，結果顯示在兩種狀態下有很大的差異，因此可以藉此辨別 DNA 是處於何種狀態。從文獻中我們得知 THz 波段的研究在各領域的重要性越來越重要。

1.2 THz 成像的特性

THz 成像的潛力主要來自於幾個特性，THz 可穿透大多數乾燥、非導電、以及無極化的物體，如紙、紙版、和無極化性的有機材料。在介電材料方面，吸收率是由光學聲子所主宰的，且與材料的極化及光學聲子共振有關。令一方面，金屬類對於 THz 是完全不透明的，

以及極化液體如水，在此波段的吸收率相當大。在氣體方面，大部分的極化分子在此波段有非常尖銳且極強的吸收譜線，這可反映出這些分子獨特的旋轉或 ro-vibration 吸收光譜。

對 THz 成像，waveform 裡面包含了豐富的資訊我們可以將所想要的資訊取出來並且即時顯示。舉個簡單的例子，我們可以使用數位信號處理器(digital signal processor ,DSP) 將時域波形做傅利葉轉換之後再將需要波段積分，如此我們可以得到不同點 THz 穿透或反射的功率，也可尋找峰值或是峰值的位置來做圖。另外一個簡單但重要的方法是時間解析，如找穿過介質之後的時間延遲，這是一個很好的判別厚度變化的方法，或者是在反射式 THz 成像中決定未知物體的位置[2]。



THz 成像之應用

THz成像大多數是用來對水分含量做分析，由於生物樣品皆含有水分，不同的組織水分含量也不同，故我們可以透過檢查水分含量來觀察組織之不同部分。THz 成像另外一個應用的方面是氣體分子的偵測，這應用需要頻率分佈很寬的輻射源才行。很多氣體分子在 mid-infrared 波段擁有獨特的吸收光譜，它們在此波段中會針對特定頻率來吸收，故我們可以透過觀察氣體吸收光譜來判斷這是何種氣體。

用這特性我們可以做一個即時的氣體監視系統。對於極化性較弱的分子如CO及CO₂在THz波段吸收量很小，因此並無法做很清楚的測定。然而對於極化很強的分子如HCl在此波段有強烈的吸收光譜，故我們可以用此系統來判斷是否含有HCl分子。

THz 成像也可用在包裹檢查或品質管制的方面上，事實上很多包裹的材料像是塑膠或卡紙對 THz 輻射而言都是透明的。在工業上這本來是使用 X-Ray 來做的，THz 成像可在這方面取代 X-Ray 因為 THz 並不會對健康或安全造成影響，其光子能量非常低且強度很弱，尤其是如果我們將飛秒雷射以光纖接到系統的話，我們就不需考慮雷射的安全。THz 成像對水分的敏感在這方面的應用是很大的，雖然說目前有不少方式可以檢查水分含量，但是只有以 THz 成像才可穿過塑膠或卡紙包裝來檢查，故我們可以用來做已包裝好的食品檢查[2]。

第二章 THz 的產生與偵測

2.1 THz 的產生

THz 輻射的產生機制主要有兩種，第一種是光電導模型，又稱電流脈衝模型(current surge model)，乃是由 Auston 等研究群所提出[7]。此乃光激載子受表面或外加電場加速而產生時變光電流，此時變即光電流產生偶極輻射，此在遠場電磁輻射與光電流對時間之一次微分成正比[1]。我們使用光導天線(photon conductive antenna)，將飛秒脈衝雷射光打在光導天線上面，天線電極兩端加電壓，如此便可放出 THz 輻射。此種方法的優點是可以簡單產生信號較強的 THz 信號；另外一種方式是利用光整流模型(optical rectification) 的方式[3] [8]，我們將飛秒(femto second)脈衝雷射打的非線性晶體上面，透過光整流效應我們也可以得到 THz 輻射，此種方式的優點是我們可以得到頻寬較寬的 THz 輻射。茲分別敘述如下。

光電導模型

天線的構造圖如下所示，一般天線是將金線鍍在 rd-SOS (Radiation Damaged Silicon On Sapphire) 或者是 LT-GaAs (Low Temperature Grown GaAs) 上面。有些天線會在兩條金線中間再做一些結構，為的是提高 THz 輻射產生的效率。THz 輻射的產生是當光

打在基板上面時會產生暫態自由電子電洞對，這些自由電子會受到兩端電場加速，按照電磁學理論電子在做加速度運動時，會放出輻射，

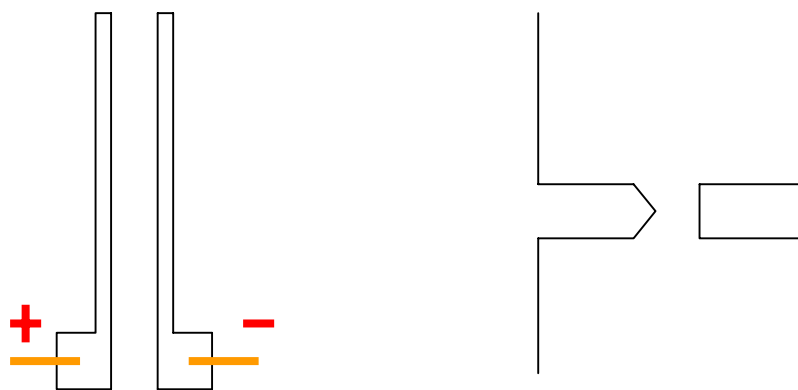


圖 2-1 THz emitter 示意圖

而這輻射就是 THz 輻射。如果我們在兩條金線中間做一些突起的結構，在這突起結構的尖端便會形成較強的電場，電子在這較強的電場中加速，所放出的 THz 輻射也較強，故可提高產生效率。

電流脈衝模型如下所述：[3]

Maxwell equations :

$$\nabla \times E = -\frac{\partial B}{\partial t} \quad (\text{Faraday's Law}) \quad (2.2.1)$$

$$\nabla \cdot E = \frac{\rho}{\varepsilon} \quad (\text{Gauss' Law}) \quad (2.2.2)$$

$$\nabla \times H = J + \frac{\partial D}{\partial t} \quad (\text{Ampere's Law}) \quad (2.2.3)$$

$$\nabla \cdot B = 0 \quad (2.2.4)$$

其中 E 為電場(electric field), B 為磁通量密度(magnetic flux), r 為電荷密度(charge density), e 為介電常數(dielectric constant), H 為磁場(magnetic field), J 為面電流密度(current density), D 為電通量密度(electric flux)。

我們可令一向量場 \vec{A} , 使得

$$\vec{B} = \vec{\nabla} \times \vec{A} \quad (2.2.5)$$

由 (2.2.1)及(2.2.5)可得

$$\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} = -\frac{\partial}{\partial t}(\vec{\nabla} \times \vec{A}) = \vec{\nabla} \times \left(-\frac{\partial \vec{A}}{\partial t}\right) \quad (2.2.6)$$

$$\Rightarrow \vec{\nabla} \times \left(\vec{E} + \frac{\partial \vec{A}}{\partial t}\right) = 0 \quad (2.2.7)$$

純量場的旋度為 0, 故我們可引入一位能 V 使得

$$-\vec{\nabla} V = \vec{E} + \frac{\partial \vec{A}}{\partial t} \quad (2.2.8)$$

$$\Rightarrow \vec{E} = -\vec{\nabla} V - \frac{\partial \vec{A}}{\partial t} \quad (2.2.9)$$

由於 $\vec{\nabla} V$ 並不貢獻與時間相關的電場, 故我們可將此忽略

$$\Rightarrow \vec{E}_{rad}(t) = -\frac{\partial}{\partial t} \vec{A}(t) \quad (2.2.10)$$

其中 $\vec{A}(t)$ 為 $\nabla^2 \vec{A} - \mu\epsilon \frac{\partial^2 \vec{A}}{\partial t^2} = -\mu\vec{J}$ 之解

$$\Rightarrow \vec{E}_{rad}(\vec{r}, t) = -\frac{1}{4\pi\epsilon_0 c^2} \frac{\partial}{\partial t} \int \frac{J_s(\vec{r}', t - \frac{|\vec{r} - \vec{r}'|}{c})}{|\vec{r} - \vec{r}'|} da' \quad (2.2.11)$$

由於觀測點距發射點甚遠, 故可使用遠場光源近似

$$|\vec{r} - \vec{r}'| = r \left(1 - \frac{\hat{n} \cdot \vec{r}'}{r} \right) \approx r \quad (2.2.12)$$

我們令光導天線是被均勻照射，故假設 \vec{J}_s 在此積分區間內為常數

$$\Rightarrow \vec{E}_{rad}(\vec{r}, t) = -\frac{1}{4\pi\epsilon_0 c^2} \frac{A}{\sqrt{x^2 + y^2 + z^2}} \frac{d}{dt} \vec{J}_s \left(t - \frac{r}{c} \right) \quad (2.2.13)$$

A 是發射天線的照射面積，我們令觀測點在軸上(i.e, x=y=0)，同時令

$t \rightarrow t - \left(\frac{z}{c} \right)$ ，我們可得

$$\vec{E}_{rad}(\vec{r}, t) \cong -\frac{1}{4\pi\epsilon_0 c^2} \frac{A}{z} \frac{d}{dt} \vec{J}_s(t) \quad (2.2.14)$$

$\vec{J}_s(t)$ 有著以下的形式

$$\vec{J}_s(t) = \frac{\sigma_s \vec{E}_b}{\frac{\sigma_s(t)\eta_0}{1+n} + 1} \quad (2.2.15)$$

n 為在 THz 波段之折射率， η_0 為在自由空間中的阻抗其值為 377Ω ，

σ_s 為導電率，其形式為

$$\sigma_s = \frac{e(1-R)}{\hbar\omega} \int_{-\infty}^t dt' m(t, t') I_{opt}(t') \exp\left[\frac{-(t-t')}{\tau_{car}} \right] \quad (2.2.16)$$

e 為單位電荷，R 為光導天線的反射率， $\hbar\omega$ 為光子能量， $m(t, t')$ 是在

時間 t' 激發在時間 t 時的載子運動性(carrier mobility)， I_{opt} 為與時間有

關的光強度， τ_{car} 是載子生命期(carrier lifetime) 我們令 carrier mobility

為常數 $m(t, t') = m$ ，以及 $\tau_{car} \rightarrow \tau$ ，我們可令入射光強度為

$$I_{opt}(t') = I_0 \exp\left(\frac{-t'^2}{\tau^2} \right) \quad (2.2.17)$$

由這些假設我們可得

$$\sigma_s = \frac{e(1-R)}{\hbar\omega} I_0 \int_{-\infty}^t dt' m \exp\left(\frac{-t'^2}{\tau^2}\right) \quad (2.2.18)$$

由 (2.2.14) , (2.2.15) , (2.2.18) 可得

$$\vec{E}_{rad}(t) = -\frac{\vec{E}_b}{4\pi\epsilon_0 c^2} \frac{A e(1-R)}{z \hbar\omega} I_0 m \exp\left(\frac{-t^2}{\tau^2}\right) \times \left[1 + \frac{\eta_0 e(1-R) I_0 m}{(n+1)\hbar\omega} \int_{-\infty}^{t/\tau} \tau \exp(-x^2) dx \right]^{-2} \quad (2.2.19)$$

所產生的 waveform 大致如圖 2-2 , 我們可以看到其分布大約是個雙極性的分布。

光整流 (optical rectification) 模型 :

此方法是利用二階非線性效應中的光整流效應來產生 THz。我們將脈衝雷射以適合角度打在非線性晶體(如 ZnTe)上, 如此便可產生 THz 輻射。

一般而言, 材料的偶極密度 P 可寫成[3]

$$P = \epsilon_0 \chi E$$

其中 ϵ_0 為介電常數

$$\chi = \chi_1 + \chi_2 E + \chi_3 |E|^2 + \chi_4 |E|^3 + \dots$$

我們可以將 P 改寫為

$$\begin{aligned} P &= \epsilon_0 \chi_1 E + \epsilon_0 \chi_2 |E|^2 + \epsilon_0 \chi_3 |E|^3 + \epsilon_0 \chi_4 |E|^4 + \dots \\ &= P^{(1)} + P^{(2)} + P^{(3)} + P^{(4)} + \dots \end{aligned}$$

若我們的入射光為兩種頻率所組成, 則我們可將入射電場寫成

$$E(t) = E_1 e^{i\omega_1 t} + E_2 e^{i\omega_2 t}$$

我們只看二階非線性效應，則我們可以得到

$\pm 2\omega_1$ 、 $\pm 2\omega_2$ 、 $\pm(\omega_1 - \omega_2)$ 、以及 0 這些頻率

這些頻率強度分別為

$$P_{2\omega_1}(t) = \chi^{(2)} E_1 \cdot E_1 e^{2i\omega_1 t}$$

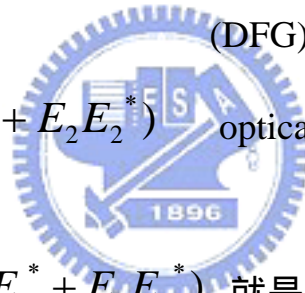
$$P_{2\omega_2}(t) = \chi^{(2)} E_2 \cdot E_2 e^{2i\omega_2 t} \quad \text{second-harmonic generation (SHG)}$$

$$P_{\omega_1 + \omega_2}(t) = \chi^{(2)} E_1 \cdot E_2 e^{i(\omega_1 + \omega_2)t} \quad \text{sum-frequency generation (SFG)}$$

$$P_{\omega_1 - \omega_2}(t) = \chi^{(2)} E_1 \cdot E_2^* e^{i(\omega_1 - \omega_2)t} \quad \text{difference-frequency generation}$$

(DFG)

$$P_0(t) = 2\chi^{(2)} (E_1 E_1^* + E_2 E_2^*) \quad \text{optical rectification (OR)}$$



其中 $P_0(t) = 2\chi^{(2)} (E_1 E_1^* + E_2 E_2^*)$ 就是我們所使用的光整流效應。

入射光源為一雷射脈衝，我們令其在頻譜分佈上的寬度為 $\Delta\omega$ ，中心波長為 ω_0 。這樣的雷射脈衝產生了一個頻率為 Ω 的極化密度 $P_i(\Omega)$ ，則 $P_i(\Omega)$ 當為所有頻率差 Ω 之總和

$$P_i(\omega) = \int_{\omega - \Delta\omega}^{\omega + \Delta\omega} \chi_{ijk}^{(2)}(\Omega; \omega' + \Omega, -\omega') E_j(\omega' + \Omega) E_k^*(\omega') d\omega'$$

其中如果入射脈衝寬度在 picosecond 範圍，則 $\Delta\omega \cong 10^{12}$ 。i,j,k 為晶體之三個軸向。若中心頻率 ω_0 遠離晶體之震盪頻率，則我們可將

$\chi_{ijk}^{(2)}(\Omega; \omega'+\Omega, -\omega')$ 視為與頻率 ω 無關。我們可以改將其表示成一平均

值
$$\chi_{ijk}^{(2)}(\Omega) = \frac{1}{\Omega} \int_{-\Omega/2}^{\Omega/2} \chi_{ijk}^{(2)}(\Omega; \omega + \Omega, -\omega) d\omega$$

則 $P(\Omega)$ 可表示為

$$P_i(\Omega) = \chi_{ijk}^{(2)}(\Omega) \int_{\omega-\Delta\omega/2}^{\omega+\Delta\omega/2} E_j(\omega'+\Omega) E_k^*(\omega') d\omega'$$

理論上，偶極(dipole)在某頻率所發出的電場，我們可表示為

$$\vec{E}(\Omega) = k^2 [\hat{n} \times \vec{p}(\Omega)] \times \hat{n} \frac{e^{ikr}}{r} + \{3\hat{n}[\hat{n} \cdot \vec{P}(\Omega)] - \vec{P}(\Omega)\} \left(\frac{1}{r^3} - \frac{ik}{r^2}\right) e^{ikr}$$

其中 $k = \frac{\Omega}{c}$ ，再加上遠場近似則

$$\vec{E}(\Omega) = -k^2 [\hat{n} \times \vec{P}(\Omega)] \times \hat{n} \frac{e^{ikr}}{r}$$

因此振幅 $E_r(\Omega)$ 可以表示成

$$E_r(\Omega) \propto \Omega^2 P(\Omega) \propto \Omega^2 \chi_{ijk}^{(2)}(\Omega) \int_{\omega-\Delta\omega/2}^{\omega+\Delta\omega/2} E_j(\omega'+\Omega) E_k^*(\omega') d\omega'$$

所發出輻射為所有頻率相加的結果

$$E_r(\Omega) \propto \bar{\chi}_{ijk}^{(2)} \int_0^\infty \Omega^2 d\Omega \int_{\omega-\Delta\omega/2}^{\omega+\Delta\omega/2} E_j(\omega'+\Omega) E_k^*(\omega') d\omega'$$

其中 $\bar{\chi}_{ijk}^{(2)}$ 為 $\chi_{ijk}^{(2)}(\Omega)$ 對各不同 Ω 的平均

以光整流效應所產生的 THz 信號如圖 2-2.a 所示，圖 2-2.b 為其傅氏轉換的結果，我們可以看到其頻率分佈大約從 0~2.5THz，最強頻率約在 1THz。不論是頻寬或是中心波長都比以天線產生要強。

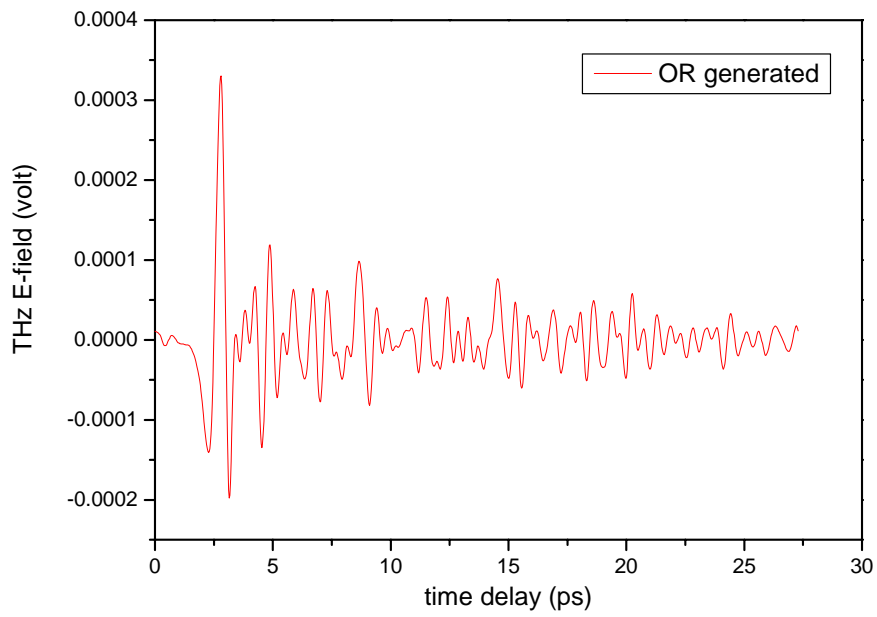


圖 2-2.a 以光整流效應產生 THz

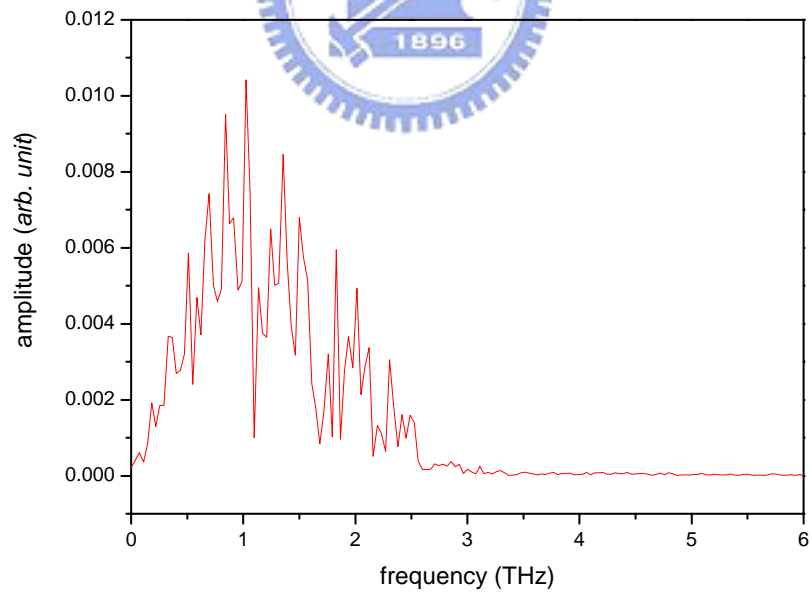


圖 2-2.b 以光整流效應產生 THz 的傅氏轉換

2.2 THz 的偵測

THz 的偵測主要可用電光取樣 (E-O sampling)、天線量測、以及 bolometer 三種方式測量。其中電光取樣以及天線量測是量測電場在時間上分布。一般的量測方法都是量某波段的能量，而這種方法則是量測信號的電場強度在時間上的分佈。如圖 2-3 所示為一 THz waveform。由於我們得到的信號是一個電場強度對時間的 waveform 而不是只有一個強度值，故我們可以從其中得到許多其他的資訊，例如特定頻率之振幅強度以及其相位資訊。

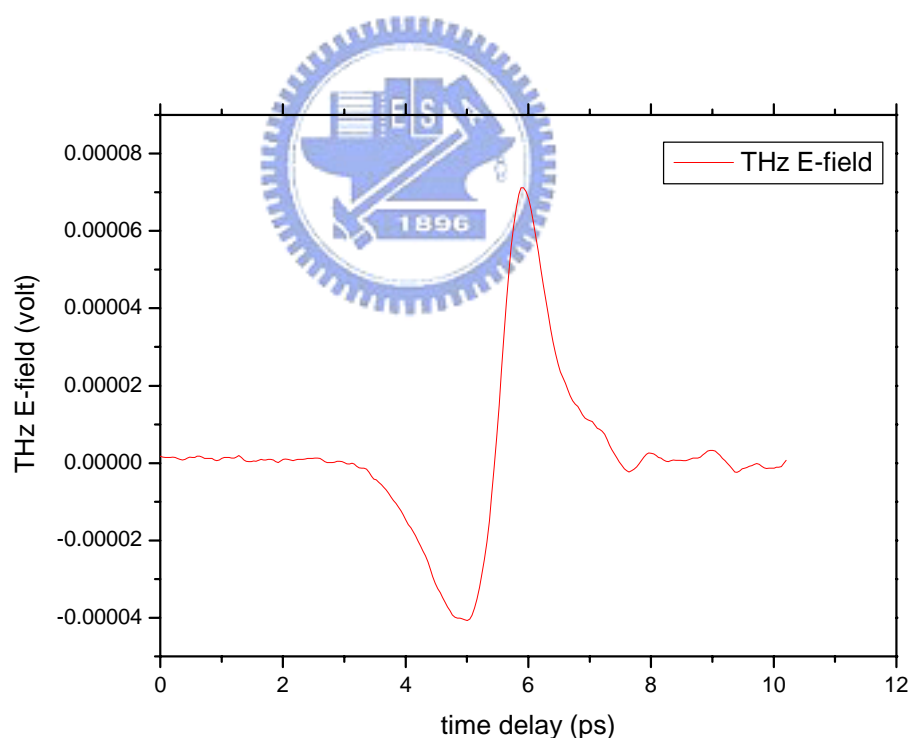


圖 2-3 掃到的 THz waveform

如圖 2-4 所示為系統架構之示意簡圖，入射脈衝雷射通過分光鏡分為兩道光，一道稱為激發光束(pump beam)用來產生 THz 信號，另

外一道稱為探測光束(probe beam)用來量測信號用。探測光束經過一個用來調整光程差的光學延遲平移台(optical delay stage)再與由激發光束所產生的信號一起打入偵測器中。由於我們所量測到的讀值正比於探測光束強度以及 THz 電場強度的乘積，我們所量到的信號正比於 $\int_{-\infty}^{\infty} E_{THz}(t)I_{opt}(t-\tau)dt$ ， $I_{opt}(t)$ 為脈衝之波包(envelope)，所以只有當雷射脈衝與信號同時打入偵測器時，我們才可量到與雷射脈衝交互作用的 THz 電場大小。我們可以調整光學延遲平移台的前後位置來控制探測光束脈衝與信號的重疊位置，如此便可量到 THz 電場強度在時域上的分布情況。

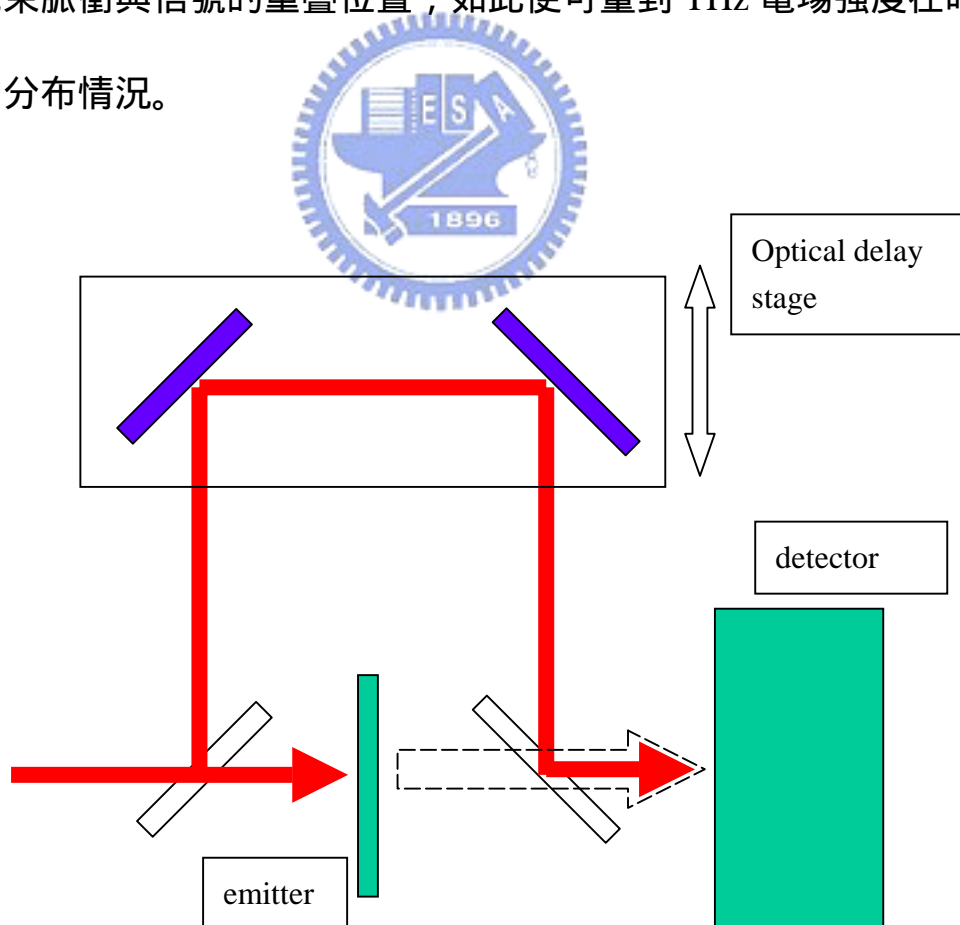


圖 2-4 系統架構示意簡圖

電光取樣

電光取樣(E-O sampling)架構圖 如圖 2-5 所示，由電光晶體(E-O crystal,如 ZnTe)、 $\lambda/4$ plate、以及一片偏極片(polarizer)所組成。當 THz 信號打在電光晶體(E-O crystal)上時，由於非線性效應的影響，便會對其快軸及慢軸的折射率產生改變。當探測光束通過電光晶體時，便會造成其垂直及水平方向有個相延遲。

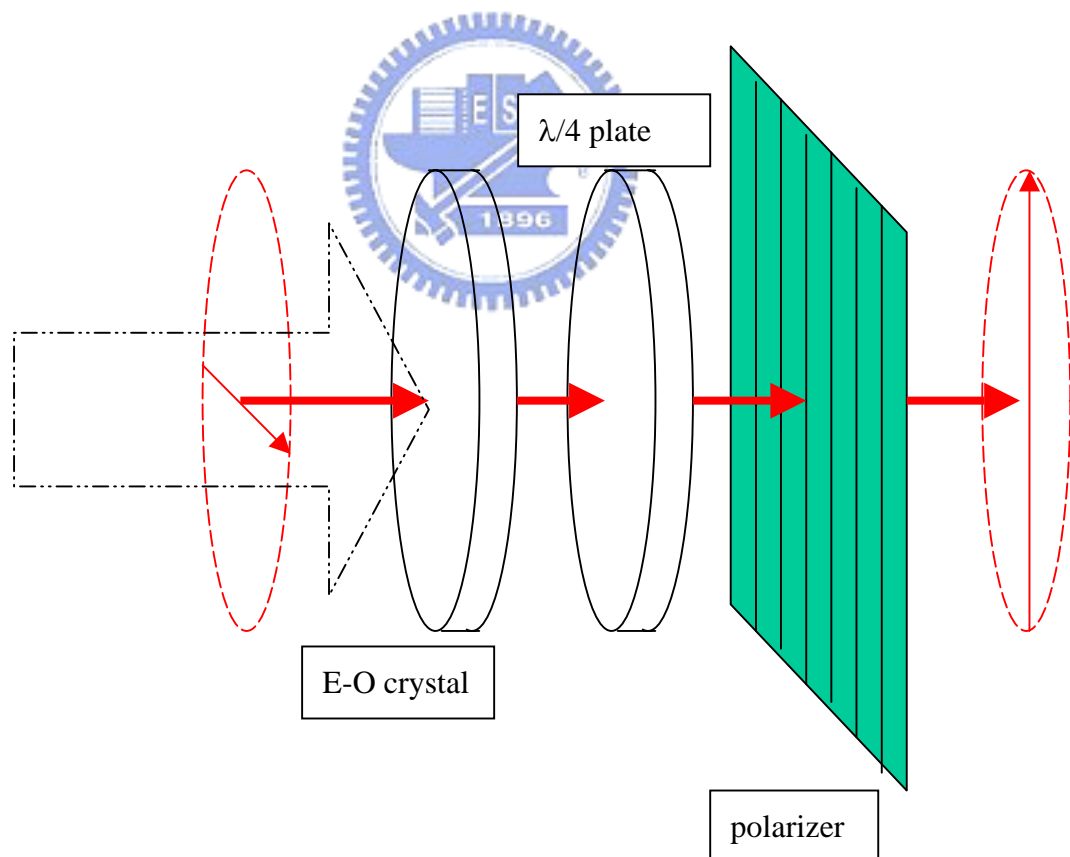


圖 2-5 電光取樣架構示意圖

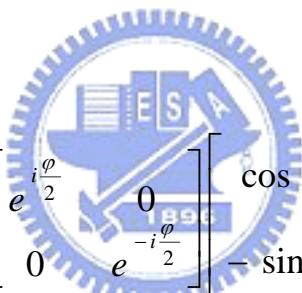
我們可以將 $\lambda/4$ plate 所造成的影響寫成以下形式

$$\begin{bmatrix} e^{i\frac{\pi}{4}} & 0 \\ 0 & e^{-i\frac{\pi}{4}} \end{bmatrix}$$

我們將其旋轉 45 度來放，故可寫成

$$\begin{bmatrix} \cos \frac{\pi}{2} & -\sin \frac{\pi}{2} \\ \sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{bmatrix} \begin{bmatrix} e^{i\frac{\pi}{4}} & 0 \\ 0 & e^{-i\frac{\pi}{4}} \end{bmatrix} \begin{bmatrix} \cos \frac{\pi}{2} & \sin \frac{\pi}{2} \\ -\sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{bmatrix}$$

假設從一開始進來處到 polarizer 之間總共造成了 φ 的 phase retardation，則我們可以寫成



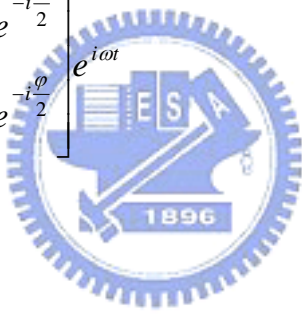
$$\begin{bmatrix} \cos \frac{\pi}{2} & -\sin \frac{\pi}{2} \\ \sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{bmatrix} \begin{bmatrix} e^{i\frac{\varphi}{2}} & 0 \\ 0 & e^{-i\frac{\varphi}{2}} \end{bmatrix} \begin{bmatrix} \cos \frac{\pi}{2} & \sin \frac{\pi}{2} \\ -\sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{bmatrix}$$

入射光若為一水平極化光，我們可將其表示為

$$\begin{bmatrix} E_0 \\ 0 \end{bmatrix} e^{i\omega t}$$

則在 polarizer 之前的電場強度為

$$E(t) = \begin{bmatrix} \cos \frac{\pi}{2} & -\sin \frac{\pi}{2} \\ \sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{bmatrix} \begin{bmatrix} e^{i\frac{\varphi}{2}} & 0 \\ 0 & e^{-i\frac{\varphi}{2}} \end{bmatrix} \begin{bmatrix} \cos \frac{\pi}{2} & \sin \frac{\pi}{2} \\ -\sin \frac{\pi}{2} & \cos \frac{\pi}{2} \end{bmatrix} \begin{bmatrix} E_0 \\ 0 \end{bmatrix} e^{i\omega t}$$

$$\begin{aligned}
&= \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} e^{i\frac{\varphi}{2}} & 0 \\ 0 & e^{-i\frac{\varphi}{2}} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} E_0 \\ 0 \end{bmatrix} e^{i\omega t} \\
&= \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} e^{i\frac{\varphi}{2}} & 0 \\ 0 & e^{-i\frac{\varphi}{2}} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} E_0 \\ -\frac{\sqrt{2}}{2} E_0 \end{bmatrix} e^{i\omega t} \\
&= \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} E_0 e^{i\frac{\varphi}{2}} \\ -\frac{\sqrt{2}}{2} E_0 e^{-i\frac{\varphi}{2}} \end{bmatrix} e^{i\omega t} \\
&= \begin{bmatrix} \frac{1}{2} E_0 e^{i\frac{\varphi}{2}} + \frac{1}{2} E_0 e^{-i\frac{\varphi}{2}} \\ \frac{1}{2} E_0 e^{i\frac{\varphi}{2}} - \frac{1}{2} E_0 e^{-i\frac{\varphi}{2}} \end{bmatrix} e^{i\omega t} \\
&= \begin{bmatrix} \cos \frac{\varphi}{2} \\ i \sin \frac{\varphi}{2} \end{bmatrix} E_0 e^{i\omega t}
\end{aligned}$$


光強度 I 為電場強度之平方

$$\begin{aligned}
I &= |E(t)|^2 \\
I &= \begin{vmatrix} \cos^2 \frac{\varphi}{2} \\ i(-i) \sin^2 \frac{\varphi}{2} \end{vmatrix} E_0^2 e^{i\omega t} e^{-i\omega t} \\
&= \begin{vmatrix} \cos^2 \frac{\varphi}{2} \\ \sin^2 \frac{\varphi}{2} \end{vmatrix} E_0^2
\end{aligned}$$

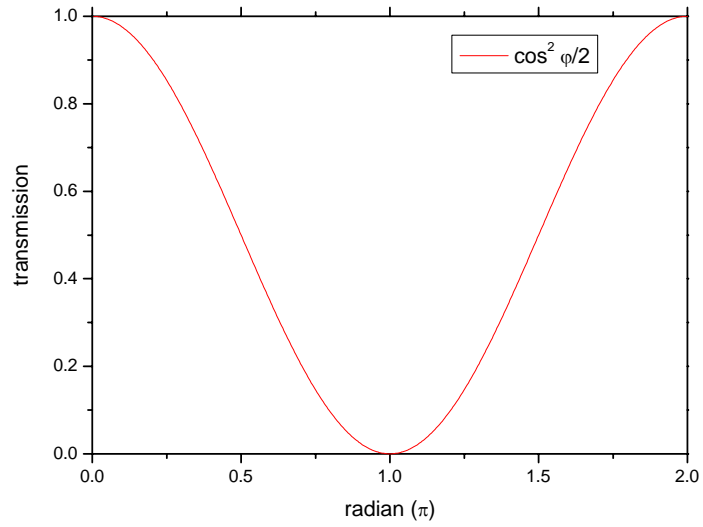


圖 2-6.a $\cos^2 \frac{\phi}{2}$ 與相位差的關係圖

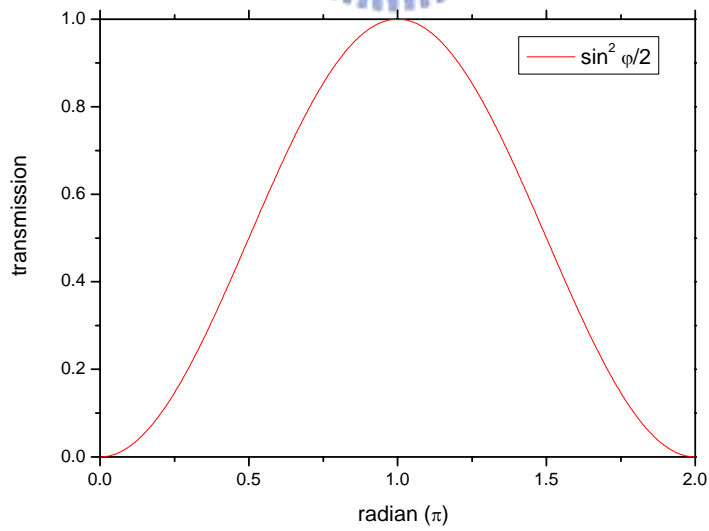


圖 2-6.b $\sin^2 \frac{\phi}{2}$ 與相位差的關係圖

如圖 2-6.a 圖 2-6.b 所示,在 $\pi/2$ 附近之 curve 幾乎呈現一線性特性,我們希望將工作點移到此處,故在中間加入一片 $\lambda/4$ plate 如此便可將操作點移到 $\pi/2$ 處。

光在通過垂直 polarizer 之後,我們便會量到

$$I = E_0^2 \sin^2 \frac{\varphi}{2}$$

其中 φ 有 $\pi/2$ 是由 $\lambda/4$ plate 來的,我們可令

$$\varphi = \frac{\pi}{2} + \varphi'$$

其中 φ' 正比於 THz 的電場強度

在 $\pi/2$ 附近我們可將 $\sin^2(\varphi/2)$ 近似為一斜率 1/2 之直線

$$I(\varphi) = E_0^2 \left(\frac{\varphi}{2} - \frac{\pi}{4} + \frac{1}{2} \right)$$

將 $\varphi = \pi/2 + \varphi'$ 代入可得到

$$I(\varphi') = E_0^2 \left(\frac{1}{2} + \frac{\varphi'}{2} \right)$$

$$I(\varphi') = I_0 \left(\frac{1}{2} + \frac{\varphi'}{2} \right)$$

由於 φ' 正比於 E_{THz} , 故我們可由 φ' 所造成的強度差得知 E_{THz} 之強度。

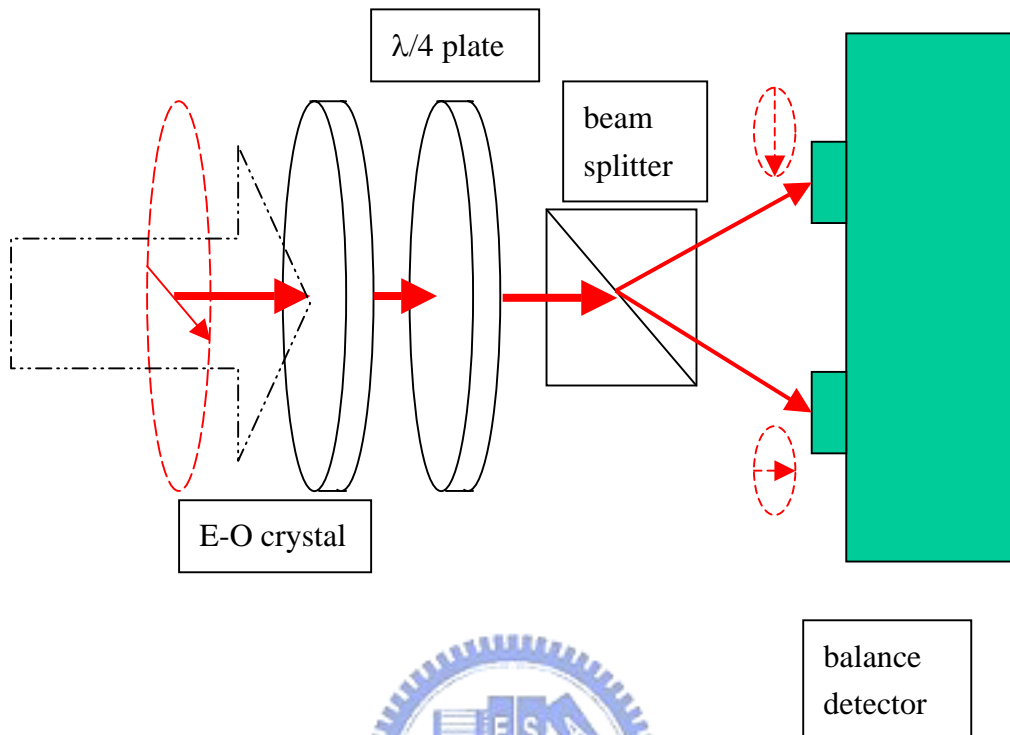


圖 2-7 以 balanced detector 來取代偏極片之電光取樣架構圖

然而在實際量測上，我們是將最後面的 polarizer 換成一個 分光鏡及平衡接收器(balanced detector)，如圖 2-7 所示。分光鏡將入射光分為水平極化與垂直極化兩道光。由前面證明垂直方向所量到強度可將其近似為

$$I_v(\varphi') = I_0 \left(\frac{1}{2} + \frac{\varphi'}{2} \right)$$

同理，我們可證明水平方向所量到強度可近似為

$$I_h(\varphi') = I_0 \left(\frac{1}{2} - \frac{\varphi'}{2} \right)$$

平衡接收器的功用是將接收到的兩端信號相減，故我們可以得到

$$I(\varphi') = I_v(\varphi') - I_h(\varphi') = I_0 \varphi'$$

這種量測方法的優點主要有兩個。第一是當垂直方向所量到的光強度 I 受到 E_{THz} 強度影響變強，則水平方向所量到的光強度就會變弱，兩者相減如此可以得到兩倍的信號強度。第二是從雷射來的雜訊會被平衡接收器相減，故可有效降低從雷射來的雜訊。



光導天線偵測

使用光導天線來量測 THz 信號的方式乃量測由飛秒脈衝雷射以及 THz 信號互相作用下所產生的電流。由於脈衝雷射所產生的暫態導電性 σ_d 非即時消失，故我們在 t_p 時間時所量到的光電流 j 其實是 THz 信號與 σ_d 在 t_p 以後的褶合積分。[3]

$$j(t_p) = \int_{t_p}^{\infty} E_{rad}(t') \sigma_d(t' - t_p) dt'$$

我們可以假設導電性 σ_d 有以下形式：

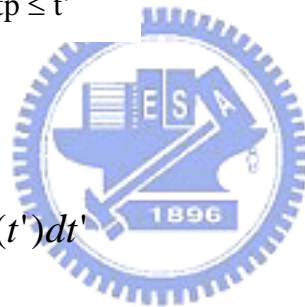
$$\sigma_d(t' - t_p) = \begin{cases} 0 & \text{if } t_p > t' \\ \frac{\sigma_0}{\tau_d} \exp\left(-\frac{t' - t_p}{\tau_d}\right) & \text{if } t_p \leq t' \end{cases}$$

其中 σ_0 為導電性的最大值。由於雷射脈衝寬度很短，故暫態導電性的增加可以被視為非常快速。我們假設 $\tau_d \gg 0$ ：

$$\sigma_d(t' - t_p) = \begin{cases} 0 & \text{if } t_p > t' \\ \frac{\sigma_0}{\tau_d} & \text{if } t_p \leq t' \end{cases}$$

將上式代入可得

$$j_p(t_p) = \frac{\sigma_0}{\tau_d} \int_{t_p}^{\infty} E_{rad}(t') dt'$$



所量到的 j_p 為 $E_{rad}(t)$ 之對時間的積分，故我們需要將量到的 j_p 微分一次才可得到真正的 $E_{rad}(t)$ 。

2.3 系統中所採用之 THz 的產生與偵測

我們的系統是以光導天線的方式來產生信號，並以電光取樣的方式來偵測信號。所使用的天線其 gap 寬度為 1.5mm，所加的電壓為 700V。ZnTe 晶體為 INGC RYS Laser systems Ltd. 所生產的，其長寬

皆為 10mm，厚度為 4mm，晶向為 $\langle 110 \rangle$ ，且為高阻抗，ZnTe 之非線性係數 r_{41} 約為 4.3，折射率約為 2.984。[13] 在系統進入前雷射功率為 195mW，經過分光鏡之後激發光束分到的功率為 177mW，在經過 chopper 以及限制通過光的 aperture 之後抵達天線的功率為 42mW，如此天線可得到光電流約 1.05mA。在探測光束方面，其強度約為 8.5mW。所得到的 waveform 如圖 2-8.a 所示，我們系統所量到的信號會在後面還有一個小信號，這是 THz 在天線中二次反射所造成的，在本圖中，我們並未將其顯示出來。我們將結果作傅氏轉換如圖 2-8.b，可以看到其最強波段約在 0.3THz 處，其頻率分布範圍約從 0 到 1.5THz。



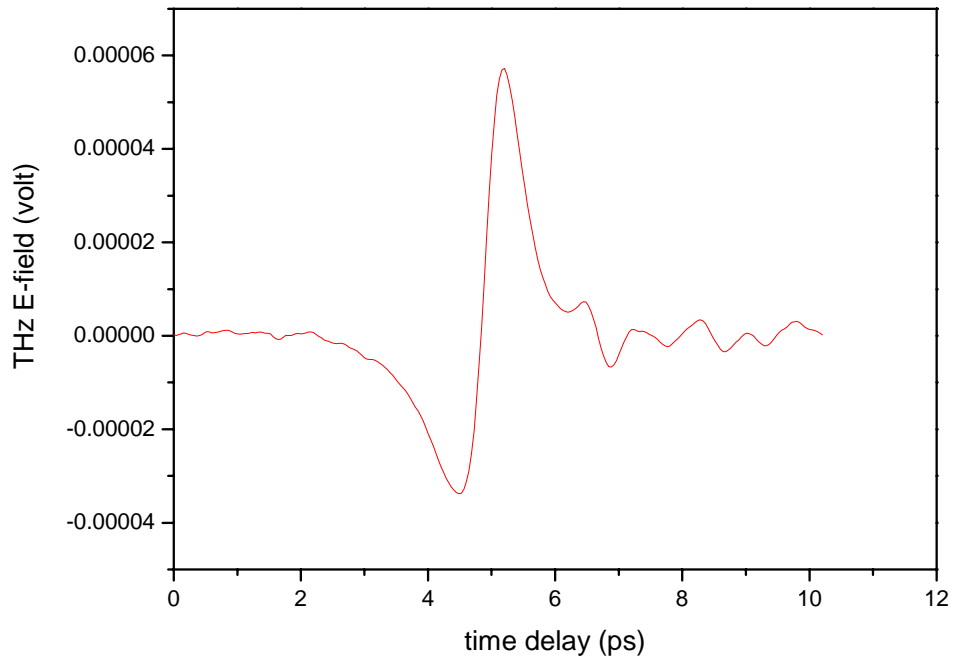


圖 2-8.a 本系統所量到之 THz waveform

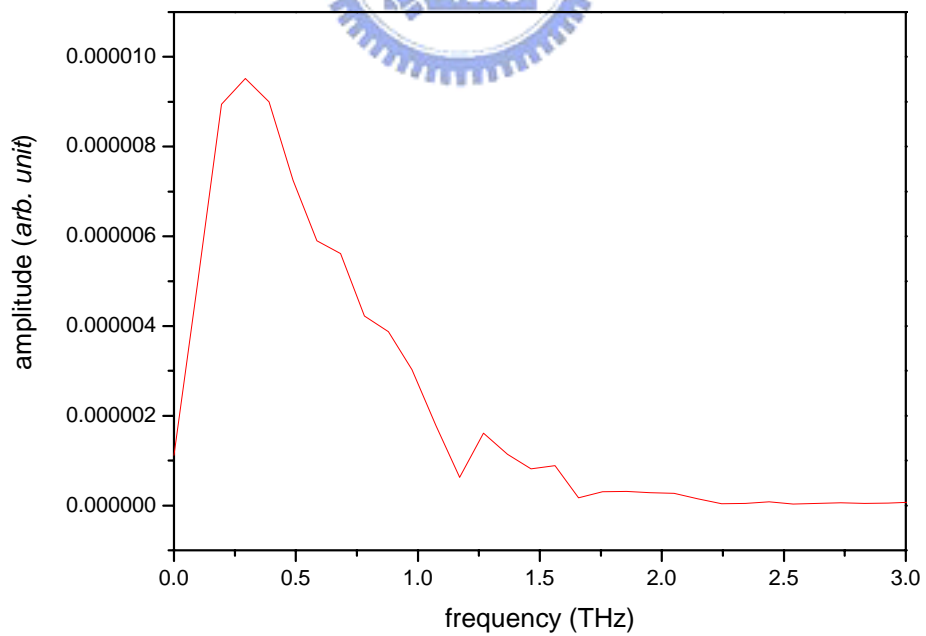
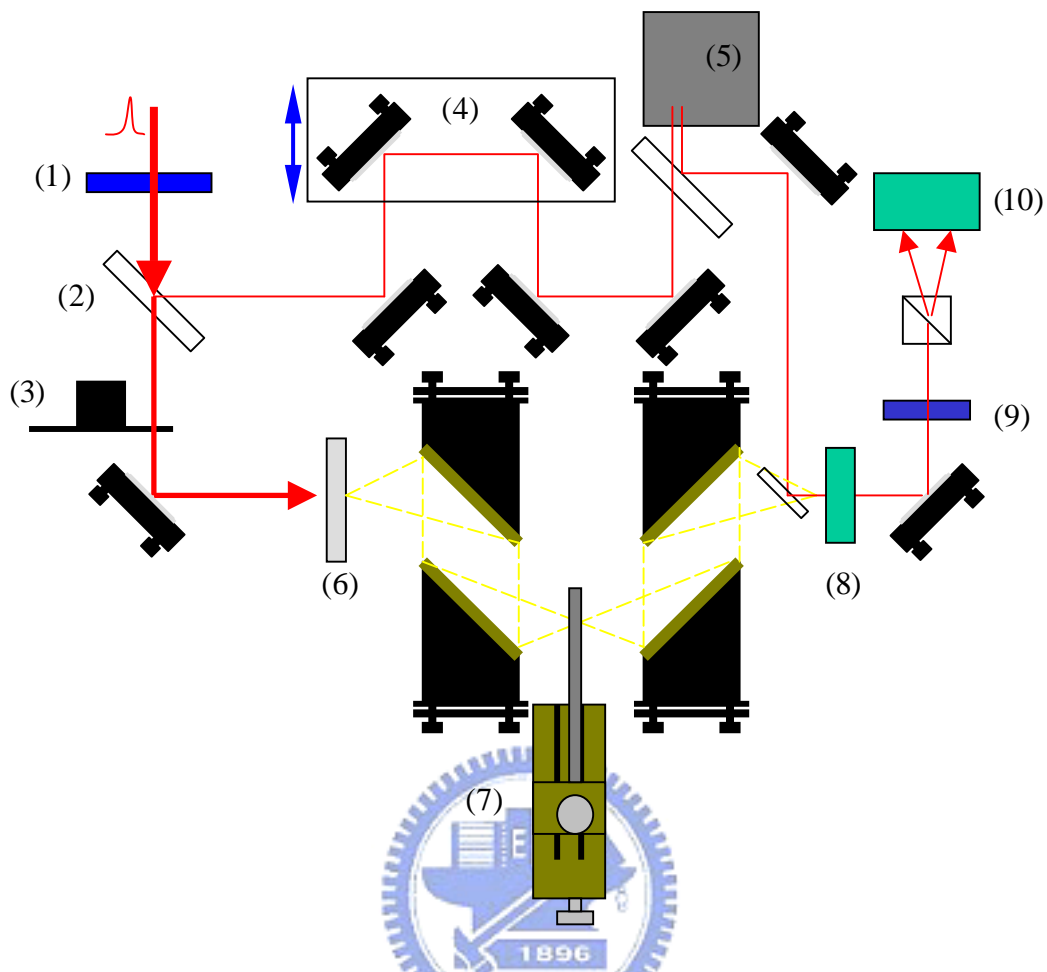


圖 2-8.b 本系統所量到 THz waveform 之傅氏轉換

第三章 THz Imaging 及系統架構

3.1 系統架構

THz 成像系統架構圖如圖 3-1 所示，我們在一個傳統的 THz waveform 量測系統上加裝一個 2D 平移台(translation stage)，我們透過移動平移台來改變量測的點，如此可以得到一個三維的資料陣列。脈衝雷射光出來時為垂直極化，打進系統中首先通過一面 $\lambda/2$ plate 將其旋轉為水平極化光。然後再通過分光鏡將其分為兩道光 (about 1:10)，一道為 probe beam，另一道為 pump beam。Probe beam 先通過一個由步進馬達所構成的光學延遲，再通過另外一台上裝有 retro reflector 的光學延遲線(ODL, optical delay line) 後面那台可以提供較快速的光學延遲，是用來做快速掃描(fastscan)用的。Pump beam 則先通過 chopper 之後再打在天線上面，產生的 THz 信號通過兩個 paraboloid mirror 收集再聚焦打在樣品上，通過樣品後再通過兩個 paraboloid mirror 收集再聚焦與 probe beam 一起打在 ZnTe 上面。我們的偵測方式是採用電光取樣，已在前面敘述。



- | | |
|-------------------------|----------------------|
| 1. $1/2$ plate | 8. ZnTe |
| 2. beam splitter | 9. $1/4$ plate |
| 3. chopper | 10. balance detector |
| 4. translation stage | |
| 5. optical delay stage | |
| 6. THz emitter | |
| 7. 2D translation stage | |

圖 3-1 THz Imaging 系統架構圖

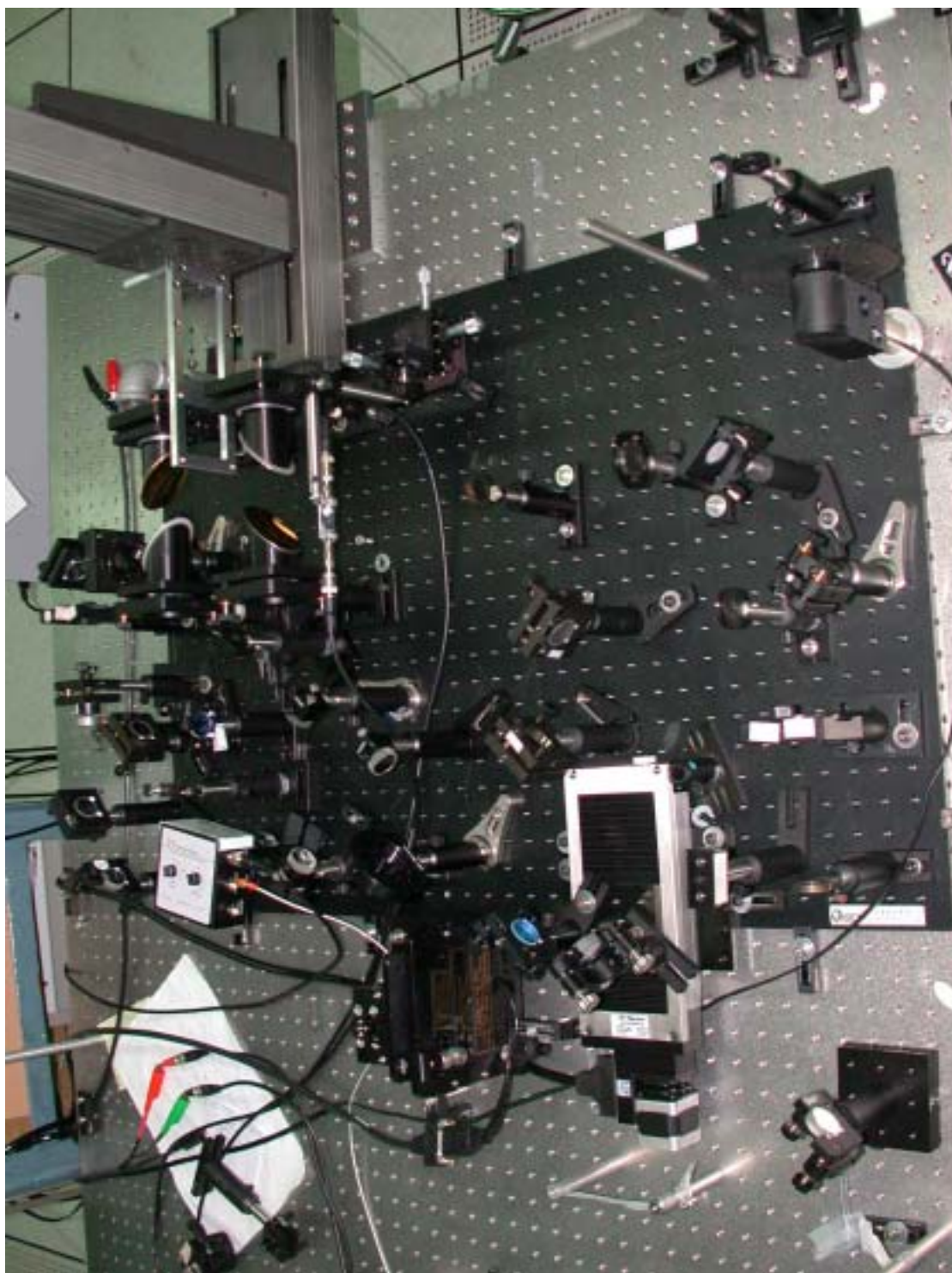


圖 3-2 THz Imaging 系統照片

項目	附註
Ti:Sapphire laser	Centered at 800 nm Vertical polarized
Chopper / Controller	PALO ALTO RESEARCH MODEL 300
發射天線	gap 1.5mm
步進馬達 / 控制器	NEWPORT ESP300
光學延遲 / 控制器	CLARK ODL150
Balanced Detector	Nirvana Model 2007
鎖相放大器 (Lock-in Amplifier)	StandFord Research SR830
數位擷取卡	ADLINK Pci9812
非線性晶體	ZnTe thickness 4mm
2D 平移台	宏惠 Unice 08TMC-2

我們所使用的光學延遲線其照片如圖 3-3.a，其極限為以 20Hz 的掃描速度提供 150ps 的時間延遲，如果需要更快的掃描速度則要降低掃描長度。其推動方式如圖 3-3.b 所示，一個可旋轉的機輪(C)被固定在一個由電信號所控制的轉軸上，為降低慣性這機輪的中心是挖空的。然

後利用彈簧片連結著一個可滑動的桿子(S)，前面裝了一個 retro reflector (R)用以反射入射光。當這機輪移動便會推動這桿子前的 retro reflector，如此便可提供光程差[10][11]。



圖 3-3.a CLARK ODL-150

取自 <http://www.cmxr.com/Scientific/ODL150.htm>

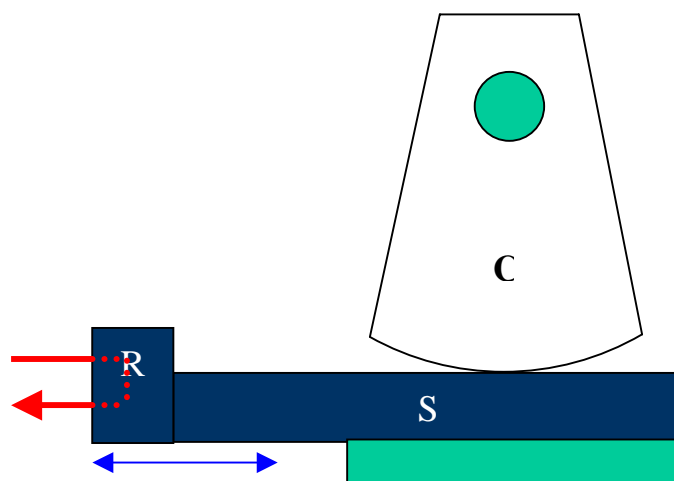


圖 3-3.b 光學延遲線推動示意圖

所量到的信號，我們可以視需求以不同的成像方式來做圖，一般而言我們可使用以下幾種成像方法：

3.2 固定時間延遲峰值成像

此種方法最為簡單快速。我們在量測時直接將步進馬達移至信號峰值所在處，再量取其各點之峰值成像。由於我們只有量測其穿透峰值大小，故無法像其他成像法可儲存大量資訊。然而由於此法的量測速度較快，故可應用在快速的檢查上面。以圖 3-4 為例是新台幣 100 元及 1000 元的浮水印。我們掃 NT. 100 所使用的解析度為每點間距 300 micrometer，所掃的範圍是 24.9mm × 24.9mm，總共 83 × 83 個像素，我們可以清楚的看到梅花浮水印以及 100 元字樣。在 NT.1000 的圖中所使用的解析度為每點 500 micrometer，所掃範圍是 35mm × 28.5mm，總共 70 × 57 個像素。掃這樣一張圖所花時間約為兩至三小時。



24.9mm

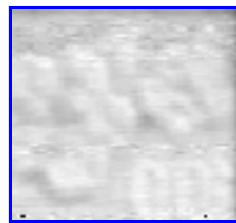
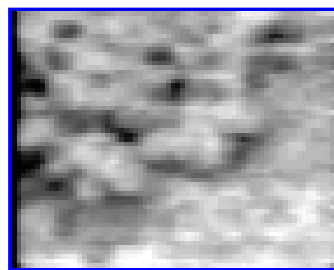


圖 3-4.a 以 THz Imaging 掃新台幣 100 元之浮水印



28.5 mm




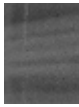
35 mm

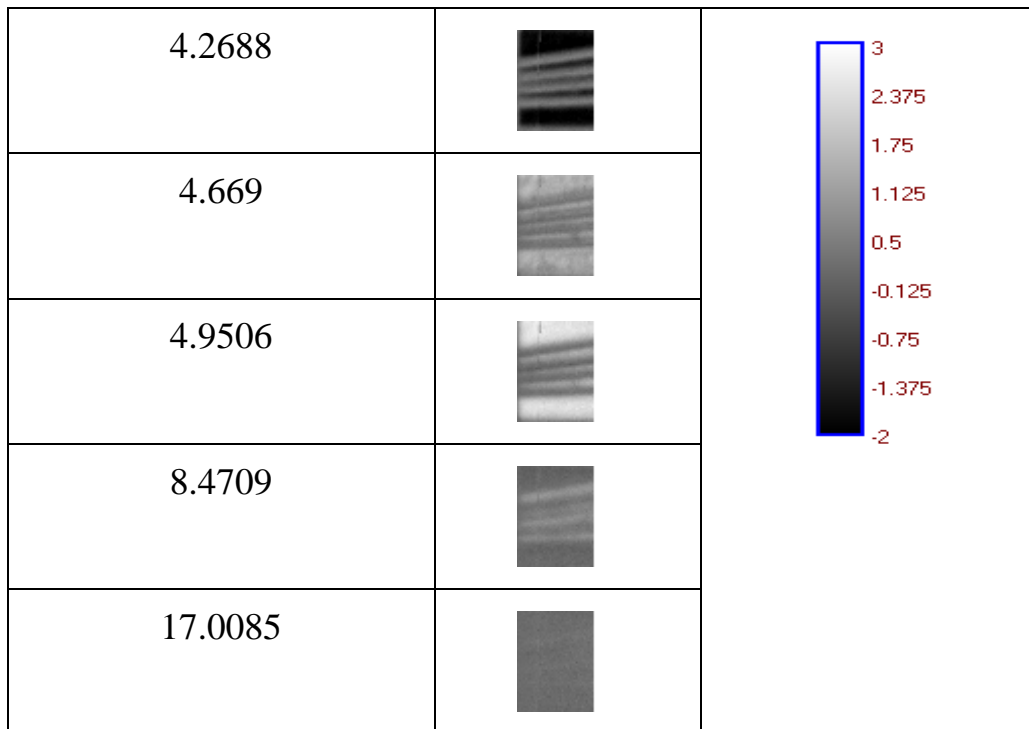
圖 3-4.b 以 THz Imaging 掃新台幣 1000 元之浮水印

3.3 不同時間延遲 電場值成像

這種方法是在不同時間延遲下，取出各點的電場值來作圖。由於時間延遲會受到樣品的折射率係數以及厚度所影響，故此種方法可以讓我們分別樣品厚度或者其折射率之變化。如下圖將一個火柴盒放了四根火柴然後在封起來去掃描。總共掃描範圍為 38×50 像素， $1.9\text{cm} \times 2.5\text{cm}$ ，每像素間距為 $500\mu\text{m}$ ，總共 1900 個像素，每個 THz waveform 掃描 256 點。所花時間為 65 小時，約兩到三天的時間。得到其電場強度對不同時間延遲輸出圖如下：



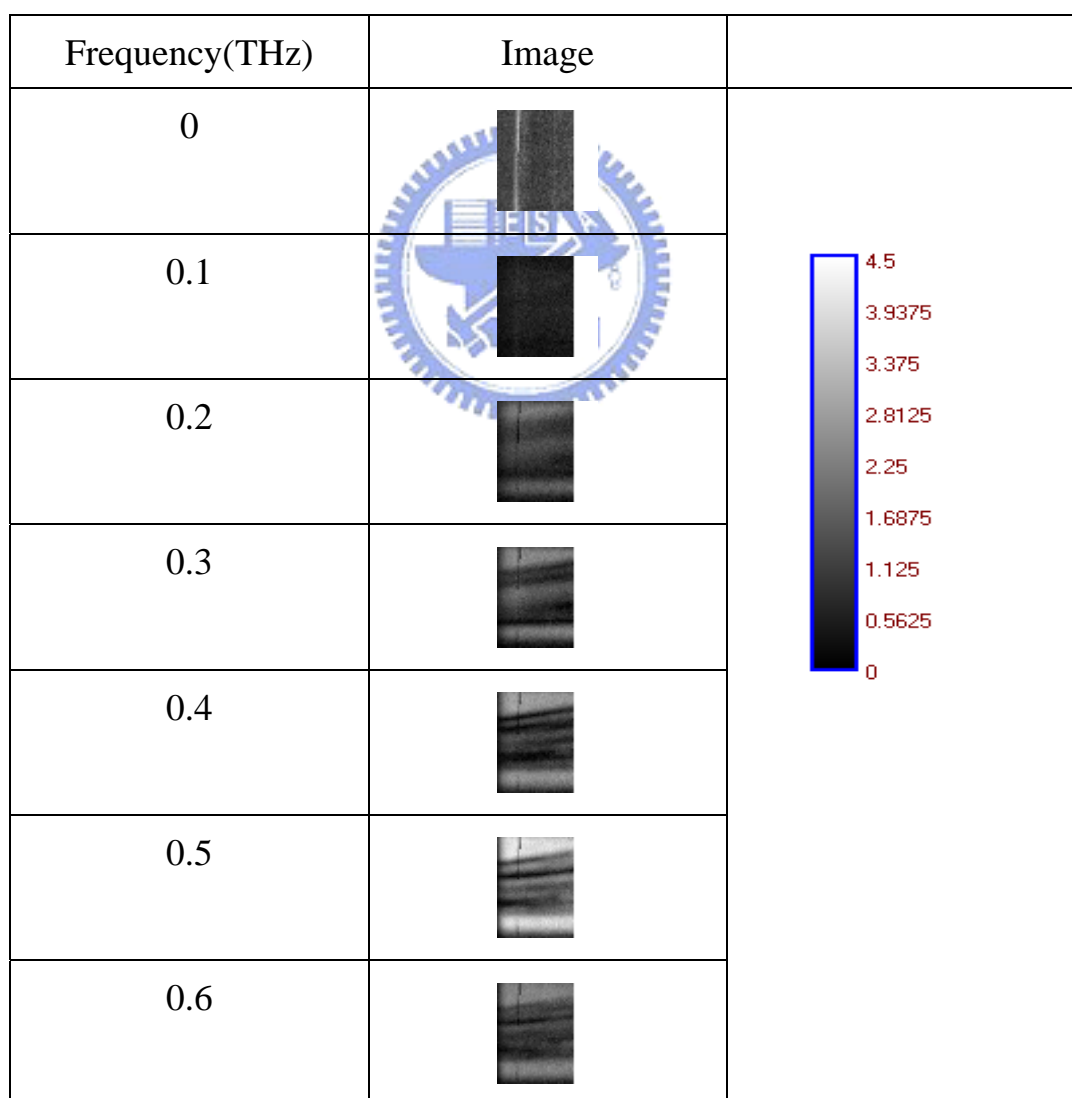
Time delay (ps)	Image	
0		
3.4017		

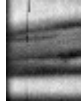
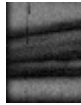
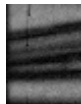


我們可以看到在不同的時間延遲中背景與火柴的明暗都不同，有時火柴較亮背景較暗，有時卻相反，這是由於不同時間延遲下，電場強度有時為正有時為負的緣故。以圖 2-2 為例，當某時間延遲背景的电場強度為負值時，而火柴由於會有散射、吸收、以及厚度所造成的像位移，它在這時間延遲下的電場值會比背景的电場強度大(或較小的負值)，我們就會看到背景為黑，火柴為白的圖像。反之，若所在時間延遲背景的电場強度為正值時，火柴的电場強度比背景的小，我們就會看到背景為白，火柴為黑的圖像。雖然火柴盒是不透明的，但是我們用 THz 仍然可以看到裡面的火柴，這驗證了 THz 高穿透性的特性。

3.4 頻域振幅作圖

我們所掃到的 THz waveform，我們可以將其用傅氏轉換轉成頻域上振幅及相位的資訊，我們可以將各個頻率的振幅取出來作圖。這種成像方式可以讓我們看樣品的吸收光譜，例如樣品的某部分對某頻率會吸收，而其他部分不會，則我們就可以在成像上看到不同。或是樣品的吸收、散射、不同厚度等原因造成穿透強度的不同。以下同樣是對上個火柴盒的掃描只是換成對頻域振幅作圖。



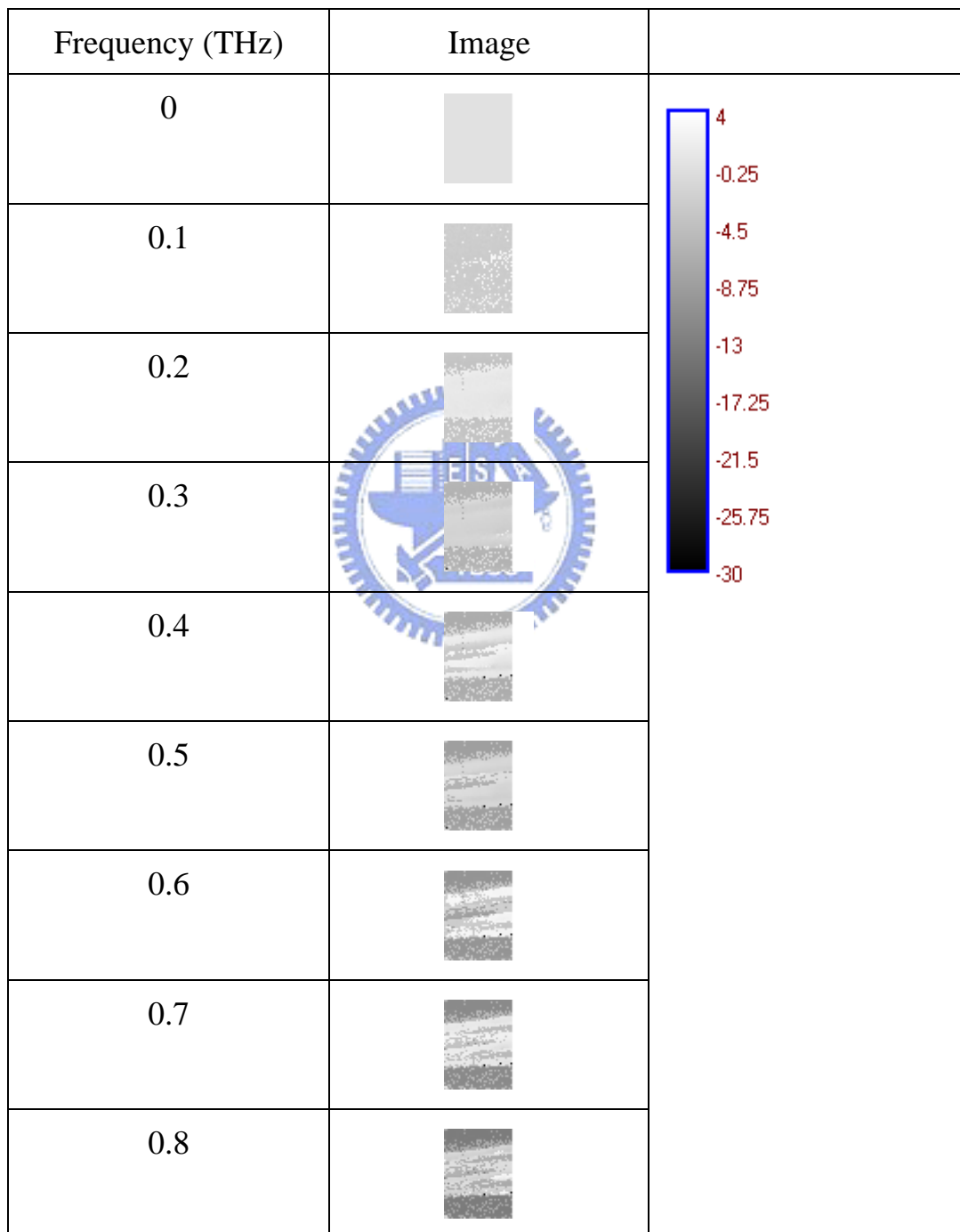
0.7		
0.8		
0.9		



從上面的結果我們可以看到雖然我們所產生的 THz 信號的最強頻率是在 0.3 THz 附近，然而在低頻部分我們看到的影像相當模糊，高頻部分強度雖弱但反而可以清晰看到有四根火柴。按照基本光學理論，一道光束聚焦最小只能聚焦到波長附近的大小，這被視作為光學的基本限制。在低頻的部分波長較長，聚焦的焦點也較大，故對較小的結構並無法清晰的成像。

3.5 頻域相位作圖

將取到的 waveform 用傅氏轉換轉成頻率，再對各頻率的相位做圖。相位做圖同樣也是被樣品的折射率以及厚度所影響，但與時域作圖法不同在於，時域作圖法所得到的圖是一個混合的結果，它並無法分辨圖像的變化是由哪個頻率所造成的，它會將較模糊的低頻以及較清晰的高頻結果混合在一起。它也可以將 THz 吸收、散射的結果反映在其中。而相位作圖法可以看出針對不同頻率的時間

延遲或折射率不同，它不會將各種頻率的結果混合在一起，也不會受到 THz 的吸收、散射等效果的影響。以下也是拿掃描火柴盒的資料來作相位作圖。



0.9		
2.0		

在低頻的部分對比度較低，這是作圖的關係。跟振幅作圖一樣，我們也可以看到在低頻的部分，火柴的影像並不是很清楚，在高頻的部分可以清楚的看到四根火柴，且在高頻時，並不會因為強度下降而造成對比度降低。



第四章 快速掃描

4.1 THz Imaging 目前缺陷

THz 成像在目前有兩大缺陷：第一是速度。一張 256 點的 THz waveform 所需時間約為一到兩分鐘，而掃描上面的火柴盒所需時間約為三天，如果我們希望每張 THz waveform 圖的點數更多的話，所花的時間更是多。第二是解析度，由於光聚焦最小只能聚到與波長左右大小，故 THz Imaging 的解析度也被限制在幾百個 micrometer 左右。我們希望可以改進上面缺陷，掃描速度方面，我們可以改用快速掃描(fast scan)的方式。解析度方面，我們可以利用提高 THz 信號的頻率等方法。



4.2 快速掃描：

快速掃描 (Fast scan)是利用一個移動快速的光學延遲線(optical delay line,ODL)來回掃描，以很快的速度取得多個 waveform，再將所得的 waveform 作平均。[10][11]我們若分析雷射的雜訊在頻譜上的分布，則我們可以發現其雜訊的分布大約是從 0Hz 到射頻 (radio frequency)，其在 0Hz 處有個最大值，然後隨著頻率的增加約略呈 $1/f$ 的衰減。我們可以快速掃描的方式將頻寬展開至高頻處，再加上高通濾波器 (high pass filter)如此可將低頻較大的雜訊濾掉。另一方面，以

步進馬達來掃一個 waveform 的時間通常較長(1~100 min , ~1000 datapoints), 這低頻的雜訊便會在我們所掃的 waveform 上產生一些起伏, 這會使 waveform 看起來較為差。如果我們使用快速掃描的方式, 以很快的速度來取得很多組信號將其平均, 故在取得一個 waveform 的時間內, 我們可以將這低頻的雜訊視為”準直流”(quasi-dc), 其對 waveform 的影響是低頻雜訊會將信號上下平移, 這上下平移的雜訊會被平均掉, 所得到的 waveform 看起來起伏便會比較少。這種方法跟傳統使用步進馬達來量的方法來比較, 另一優點在於使用步進馬達來量取訊號所需時間與掃描的點數成正比, 當我們希望取的點數多的時候, 所需時間也會直線上升。而快速掃描這方法的優點在於我們所花的時間是用來取多個 waveform 來平均, 只要達到了所需求的 SNR(signal to noise ratio), 則取多少點都不會影響掃描的時間。故所需的點數越多, 則快速掃描的優點越明顯。如下圖所示為傳統步進馬達與快速掃描所量測 wave form 之比較。

在下圖中, 這些都是 2048 點的 THz wave form。圖 4-1.a 是以步進馬達掃的, 鎖相放大器(lock-in amplifier)的時間常數 (time constant) 是使用 300ms, 以步進馬達掃這樣一個 2048 點的 THz waveform 所需時間約是 12 分鐘, 其訊噪比約為 86 (我們定義訊噪比為尖峰值的強度除以前 500 點的平均標準差)。以下以快速掃描來掃的 THz

waveform 其鎖相放大器的時間常數為 3ms ， ODL 之來回掃描頻率為 1Hz ， 輸出位置函數(position function) 為一 symmetry 為 5% 的三角波。在圖 4-1.a 中，我們可以看到有許多較細的雜訊，這是由於較強的低頻雜訊所造成的，而在圖 4-1.b、圖 4-1.c、圖 4-1.d 中這情況已經改善很多了。我們可以看到圖 4-1.b 是以一分鐘時間來掃的已經有不錯的訊噪比了，其訊噪比約有 74，圖 4-1.c 是以兩分鐘掃的已經比以步進馬達掃的好一些了，其訊噪比約有 126。而圖 4-1.d 是以三分鐘掃出來的結果已經明顯超過以步進馬達所掃的，其訊噪比約有 148。

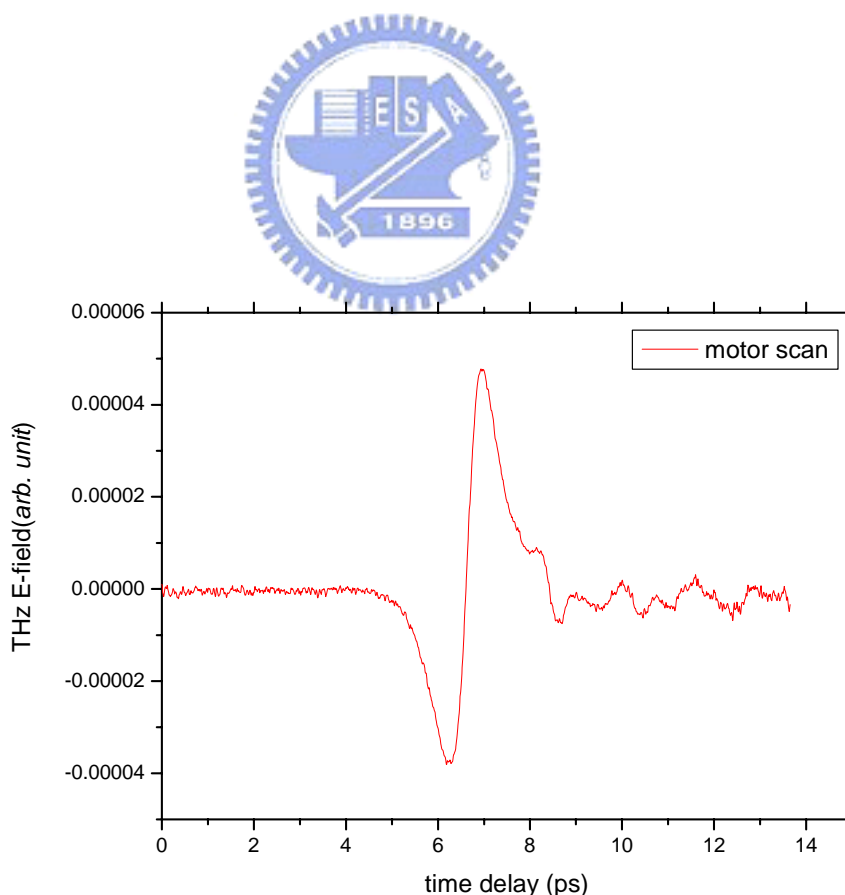


圖 4-1.a 以步進馬達來掃 THz 的 waveform

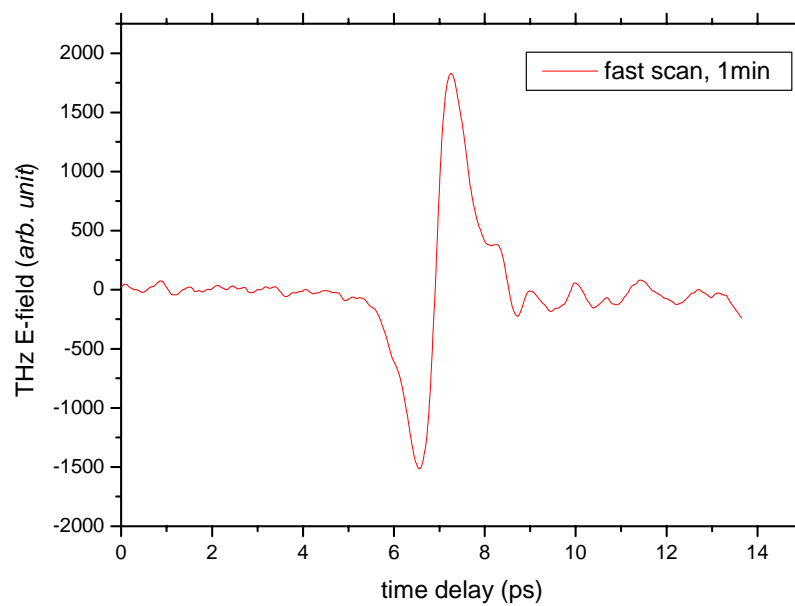


圖 4-1.b 以快速掃描的方式來掃 THz 的 waveform，掃描時間為一分鐘

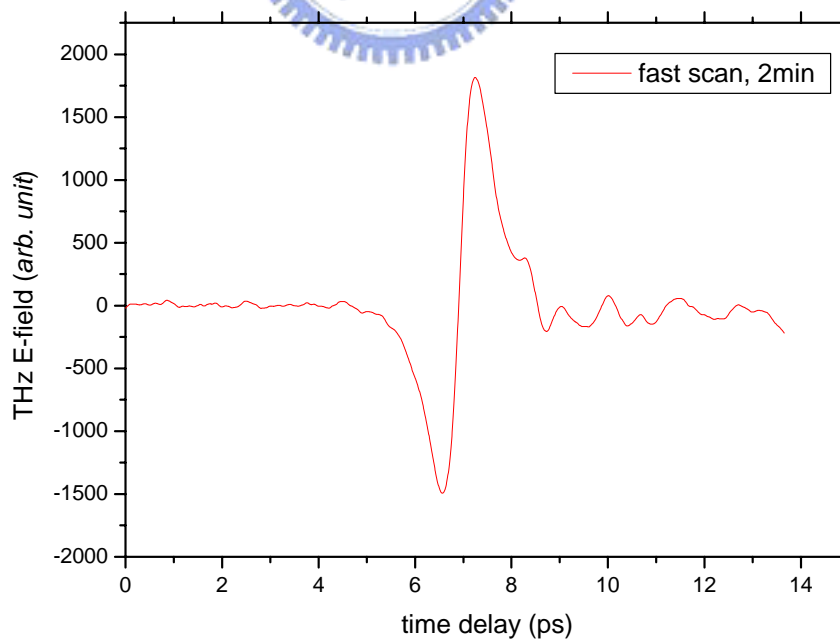


圖 4-1.c 以快速掃描的方式來掃 THz 的 waveform，掃描時間為兩分鐘

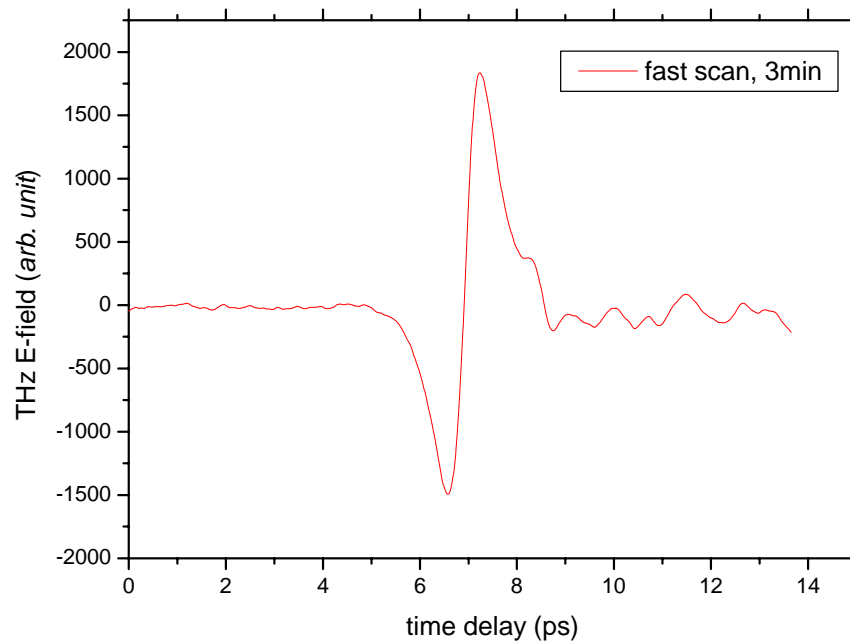


圖 4-1.d 以快速掃描的方式來掃 THz 的 waveform，掃描時間為三分鐘

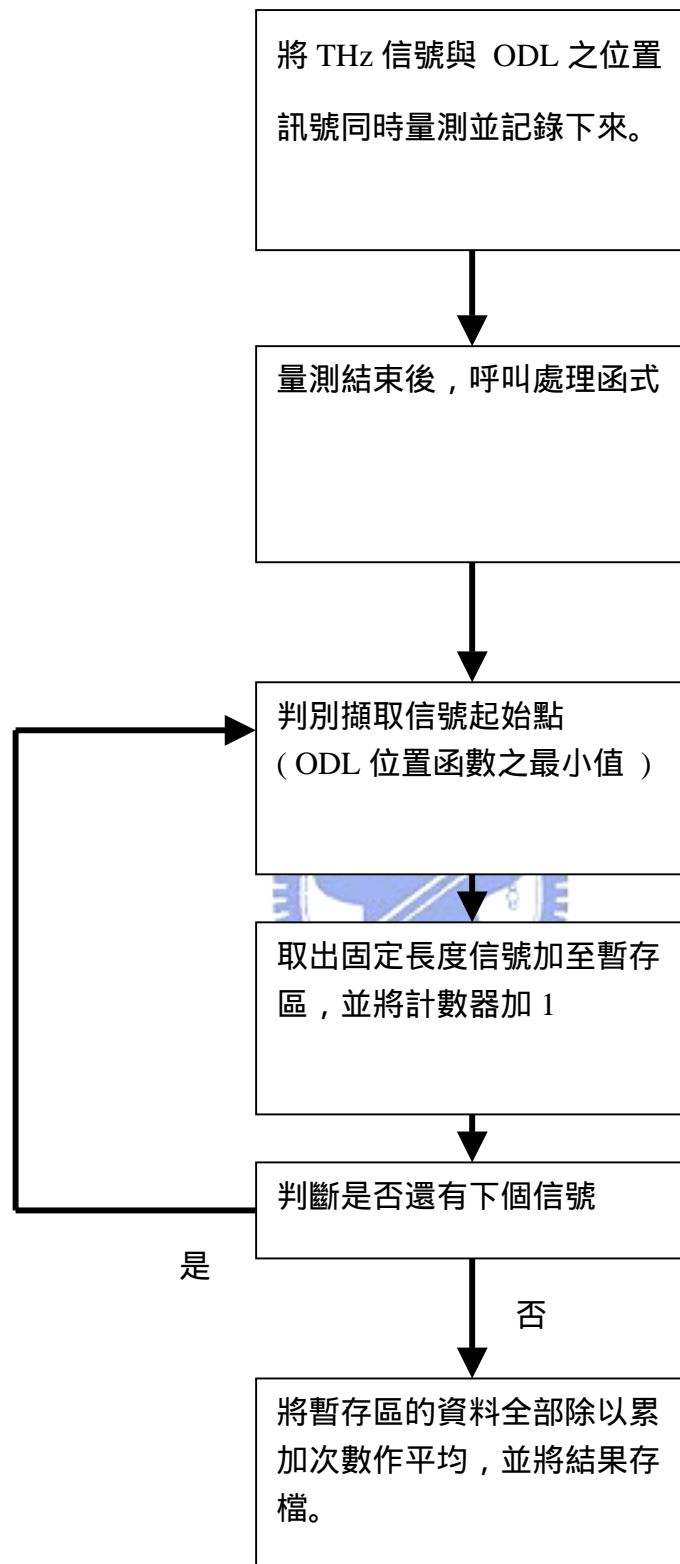


4.3 快速掃描之量取

我們所使用的光學延遲線(ODL)是以一個外界所提供的信號來驅動，我們用來驅動的信號源為一信號產生器(function generator)。這台光學延遲線的極限為 20 Hz 提供 150ps 的時間延遲(time delay)，我們將信號從鎖相放大器(lock-in amplifier)的輸出端取出來，與光學延遲線所輸出的位置函數一起接進數位擷取卡中。我們將由光學延遲線

所輸出的位置函數與從鎖相放大器所輸出的信號一起用數位擷取卡紀錄下來。之後再用程式的方法加以平均。圖 4-2.a 是從數位擷取卡所抓到的信號，粗線是從光學延遲線所傳回的位置函數，這是一個正比於光學延遲的一個信號。我們可以看到這是一個 symmetry 為 5% 的三角波，我們做平均時擷取信號的起點便是這三角波的區域最小值。而細線就是所讀到的 THz 信號。這個 waveform 是以鎖相放大器時間常數 1ms，光學延遲線掃描頻率 4Hz 所掃出來的，每個 waveform 所取的點數是 1900 點。圖 4-2.b 是圖 4-2.a 的 500 筆平均。雖然在圖 4-2.a 中我們看不出有信號的樣子，然而將其 500 筆平均之後，我們便可看到其信號。如下圖所示為作快速掃描的步驟圖





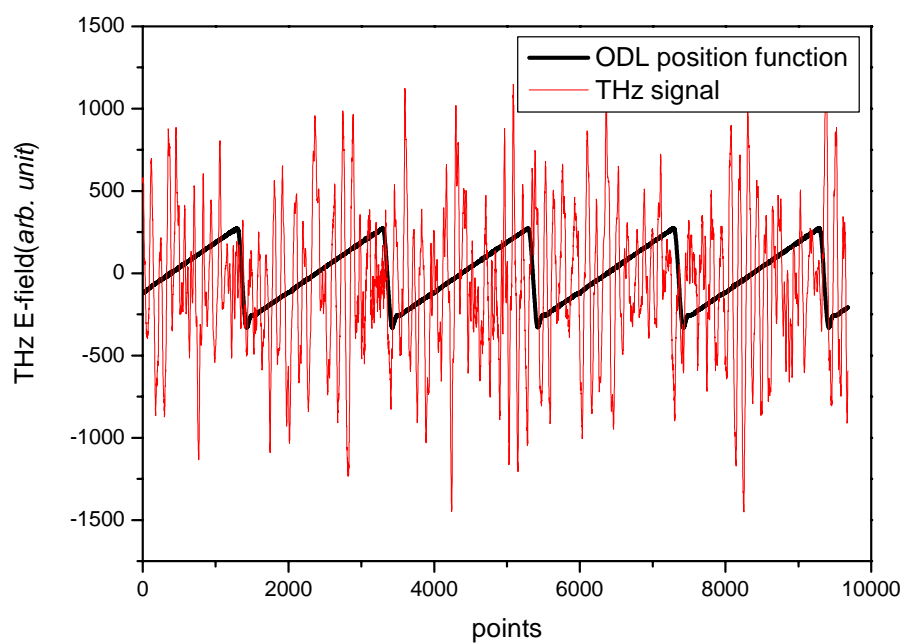


圖 4-2.a 快速掃描擷取信號的一部份

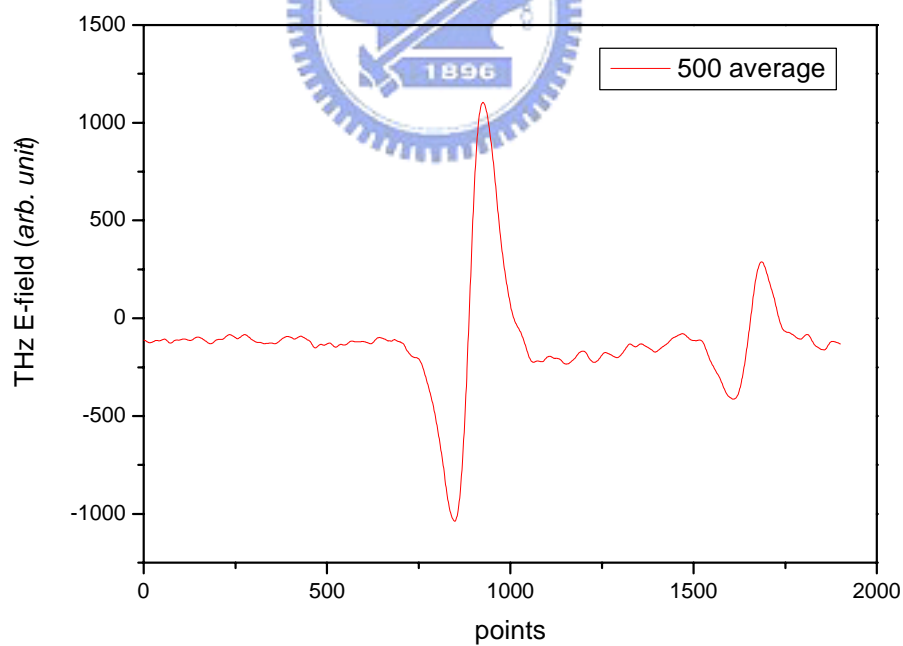


圖 4-2.b 其 500 筆數據的平均

4.4 Rescale :

由快速掃描所掃到的 THz waveform 其橫軸的單位為點數，而不是時間。我們必須要將其轉換為空間上的長度，再將其轉為時間上的延遲。我們所使用的方法是先將步進馬達停在某位置量一個 THz waveform，再將步進馬達移動到另一位置再量另一個 waveform，觀察這兩張 waveform 其信號位移了多少距離，我們便可將點數換算成距離，知道距離之後，我們便可將其轉換成時間延遲了。如下圖所示，在圖 4-3.a 中，我們將步進馬達停在 -10.8mm 處，其尖峰值是在 1246 的位置。接著將步進馬達移至 -11.4mm 處，其尖峰值位置則移動至 675 的位置。我們可以將 $600\mu\text{m}$ 換算成時間延遲，步進馬達移動 $600\mu\text{m}$ ，由於光來回一趟所走的距離是步進馬達移動距離的兩倍，故實際造成的光學延遲為 $2 \times 600\mu\text{m} = 1200\mu\text{m}$ 。將其轉換成時間延遲可以得到 $2 \times 0.0006 / 3 \times 10^8 = 4 \times 10^{-12} \text{ sec} = 4\text{ps}$ 。尖峰值位置由 1246 移至 675 共移動了 571 點，故可換算每一點的時間延遲為 $4 / 571 = 0.007005\text{ps}$ 。圖 4-3.b 為轉換好的快速掃描與步進馬達所掃的 waveform 之比較，我們可看到其幾乎完全重合，圖 4-3.c 為其傅氏轉換，我們可以看到中心波長大約是落在 0.3 THz 附近，頻率分布約為 $0 \sim 1.5\text{THz}$ 。

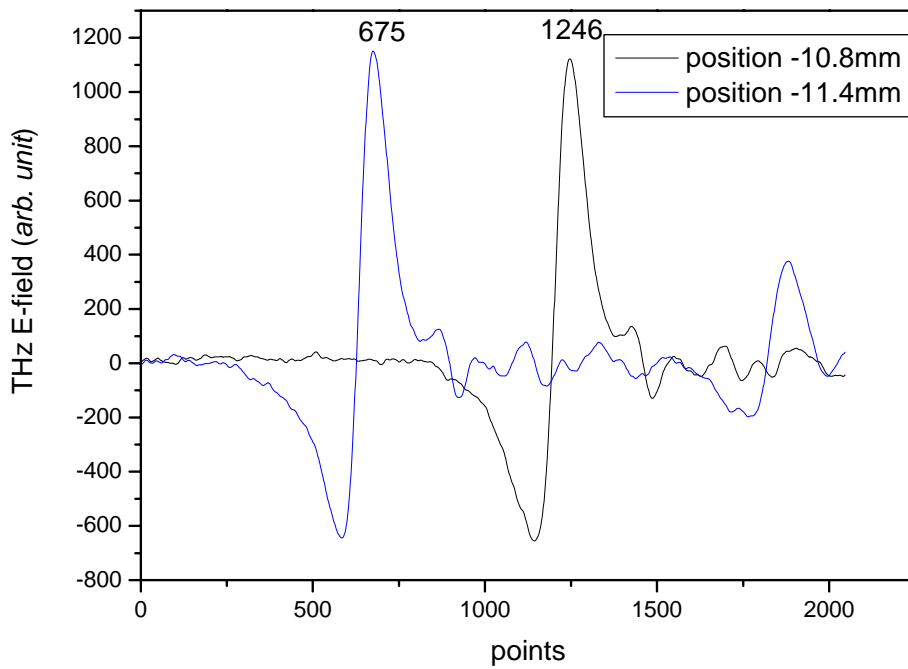


圖 4-3.a 於步進馬達位置 -10.8mm 與 -11.4mm 處所掃 waveform

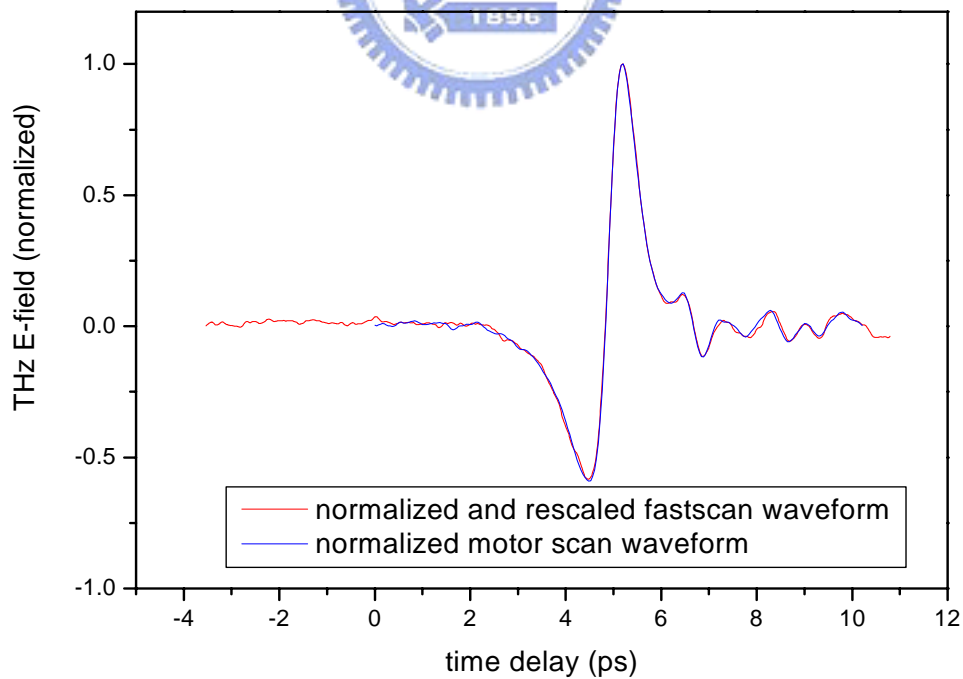


圖 4-3.b Rescale 過後的 fastscan 與 slowscanned waveform 之比較

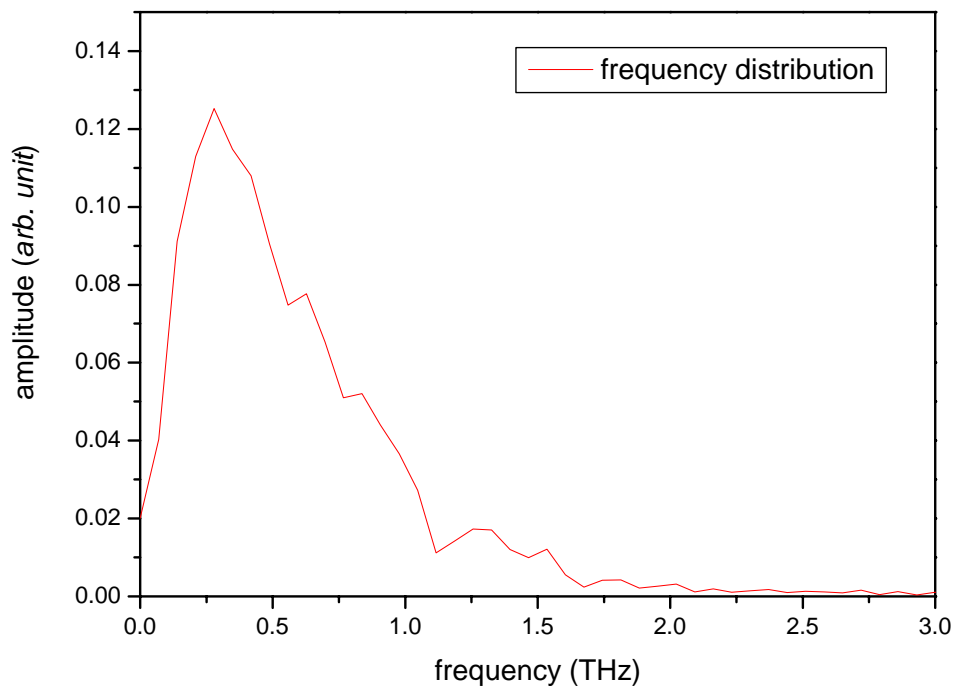


圖 4-3.c 其傅氏轉換



4.5 鎖相放大器之時間常數的選擇

在鎖相放大器(lock-in amplifier)中有個積分用的電容,用以得到更穩的信號。由於積分在數學上有著類似平均的功能,故這積分器可以將時間上某段的信號平均再輸出。這樣做會有個問題就是積分器會限制住信號變化的速度,如果信號變化的速度超過了積分器的限制,這樣信號便會失真。這失真對波形的影響主要有兩個,第一是峰值的降低,當信號到達了峰值,而鎖相放大器的反應還無法到達,而信號此時開始往下降,故鎖相放大器的讀值便無法到達信號的真正峰值。另

外一個影響是 waveform 的延展(峰值位置偏移)，當信號到達峰值時開始往下降，但鎖相放大器的反應並不會馬上跟著下降，而會繼續上升一段長度，這樣便會使得峰值的所在位值偏移。故我們要得到不失真的信號便需要在掃描速度以及失真之間找到個平衡。可能影響失真的因素有幾個，例如掃描頻率、掃描長度、信號大小 以及時間常數。在圖 4-4 中我們可以看到時間常數對於信號失真的影響，在時間常數 30ms 的圖中，我們可以看到其 curve 較為平滑，但是信號卻嚴重的失真，其信號強度減少了很多。

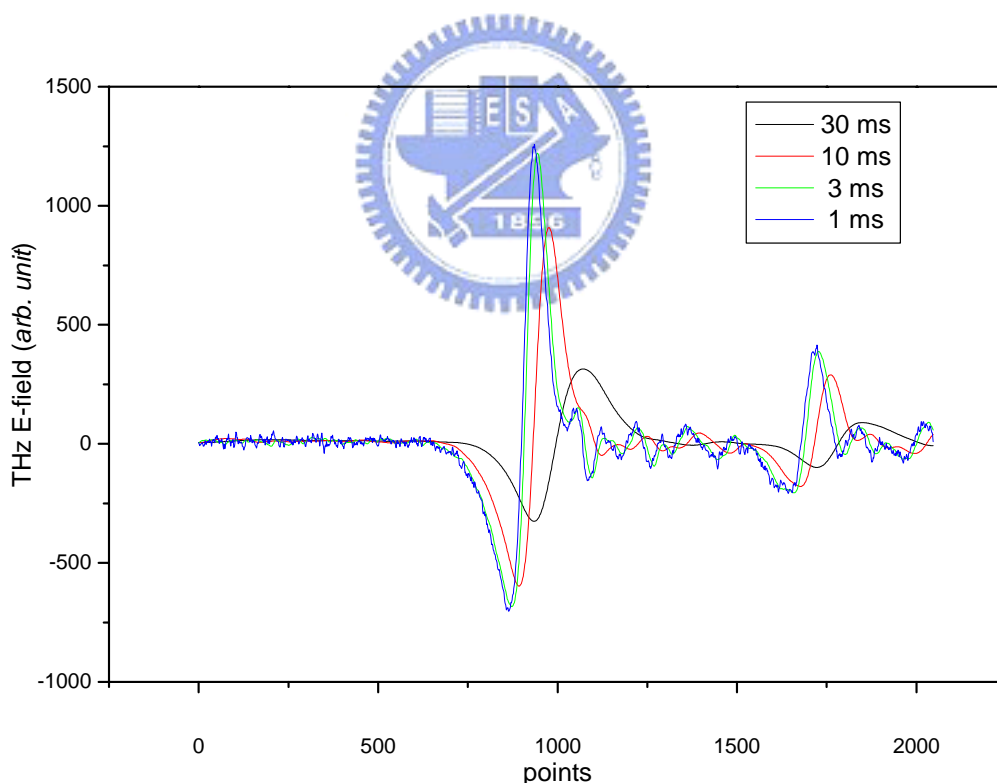


圖 4-4 鎖相放大器不同時間常數下所取得的 waveform

所以我們需要找出在不同時間常數下所能使用的最快掃描速度。如下圖所示,是以時間常數 3ms 所作的結果,掃描頻率以 0.5 Hz 以及 1 Hz 所掃的 waveform 吻合度很好,但是由掃描頻率 2 Hz 所掃的 waveform 強度開始下降。故我們想要有更快的掃描速度必須要降低鎖相放大器的時間常數。如下圖所示是我們改以時間常數 1 ms 所作的實驗,我們可以看到掃描頻率 2 Hz、3 Hz、以及 4 Hz 所掃的 waveform 重合度很好,但 5 Hz 就有些失真,故我們可以判斷在不失真的前提下,我們可以得到的最快掃描速度為 4 Hz,鎖相放大器的時間常數為 1 ms。

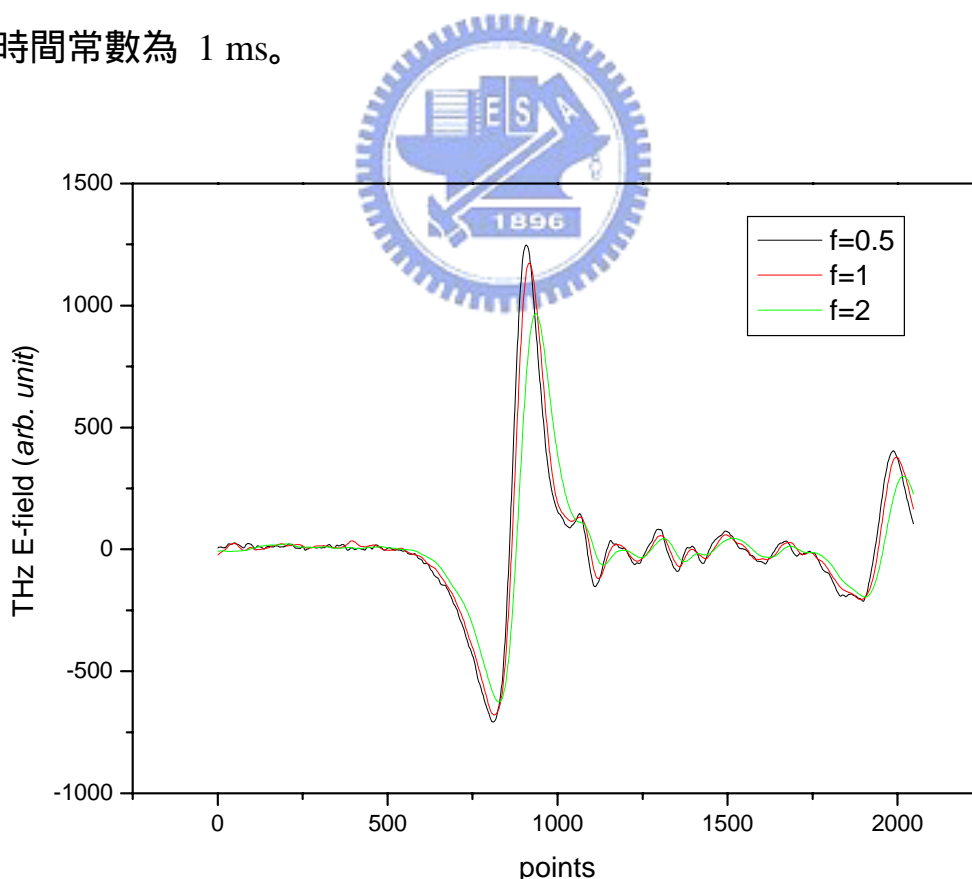


圖 4-5.a 以時間常數 3ms 下,不同掃描頻率所掃的 waveform

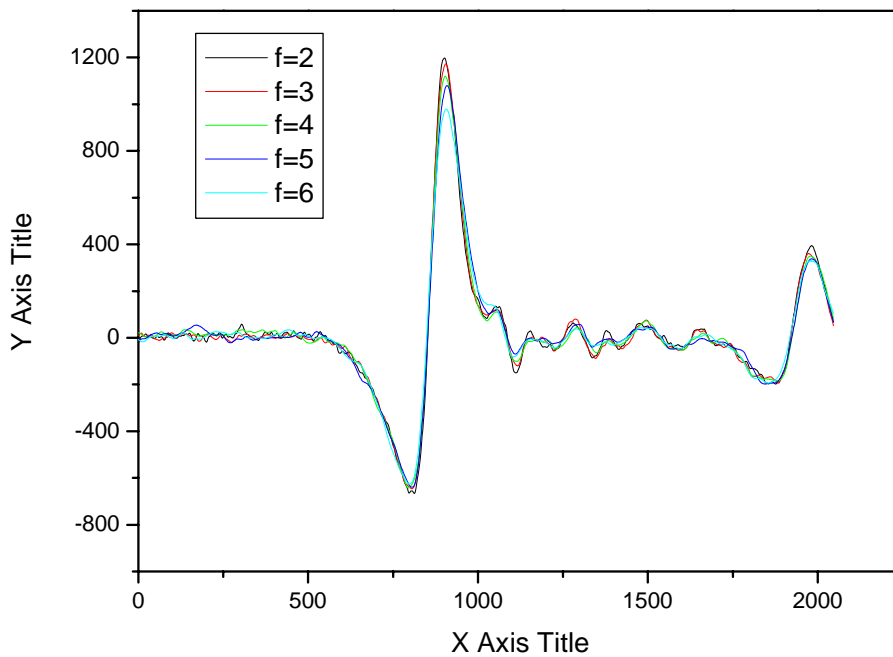


圖 4-5.a 以時間常數 1ms 下，不同掃描頻率所掃的 waveform



4.6 平均次數與訊噪比

我們將所取得的數據加以平均，如果我們平均的數據越多則我們可以獲得越好的 waveform，然而這也要花越多的時間。在本小節中，我們對系統所得到的訊噪比作測試，供未來決定掃描時間的參考。我們的實驗條件為：鎖相放大器時間常數 1ms，掃描頻率 3Hz。

我們將其中一些 waveform 展示出來，最後再以訊噪比對所花時間的圖來結束。我們定義訊噪比為最大值除以前 250 筆數據的平均標準差。

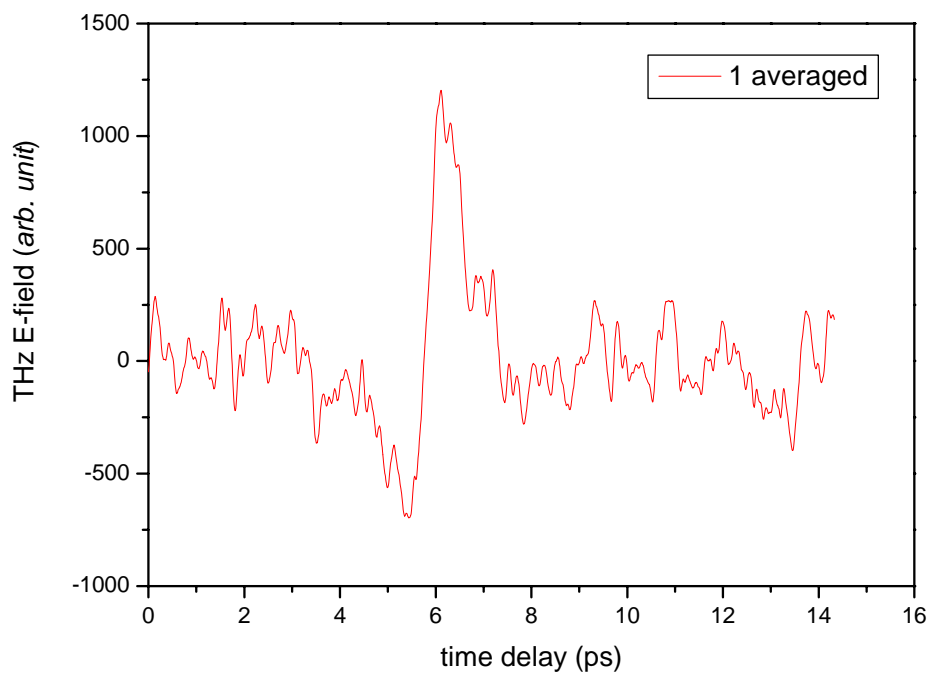


圖 4-6.a 一筆 waveform

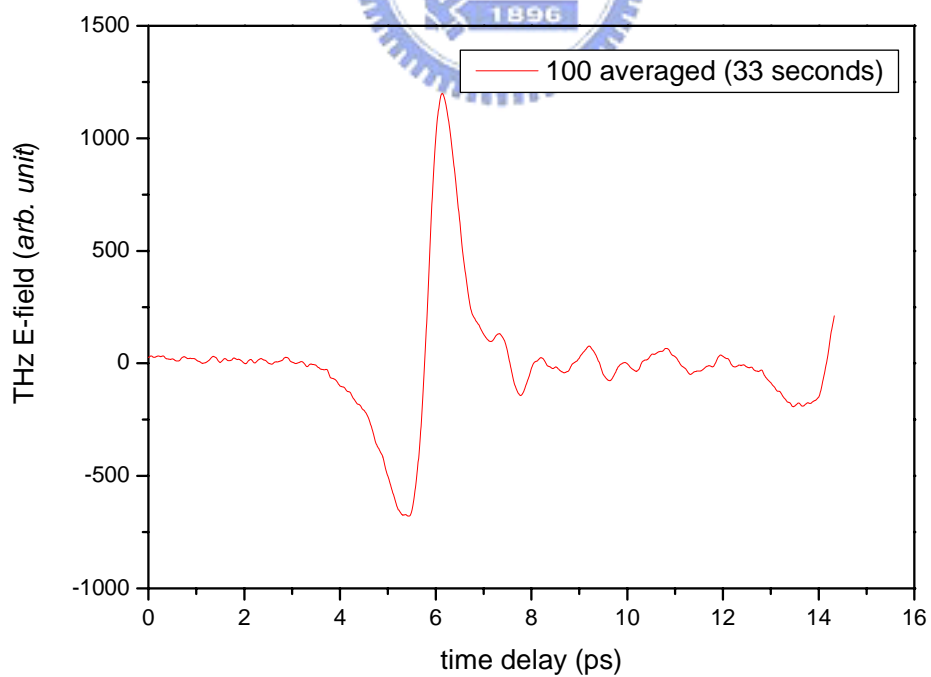


圖 4-6.b 100 筆 waveform

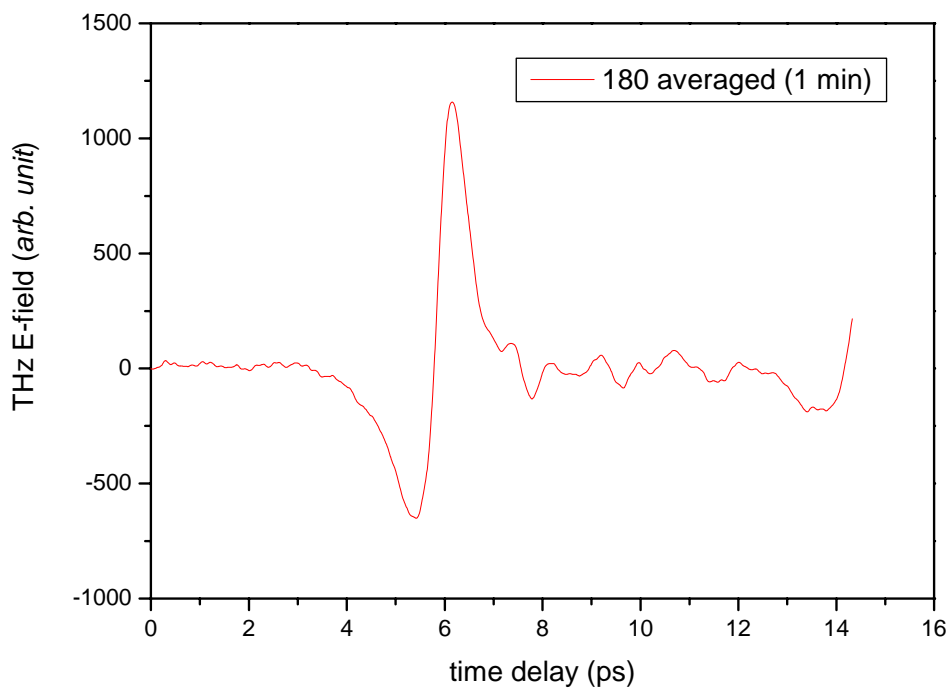


圖 4-6.c 180 筆 waveform

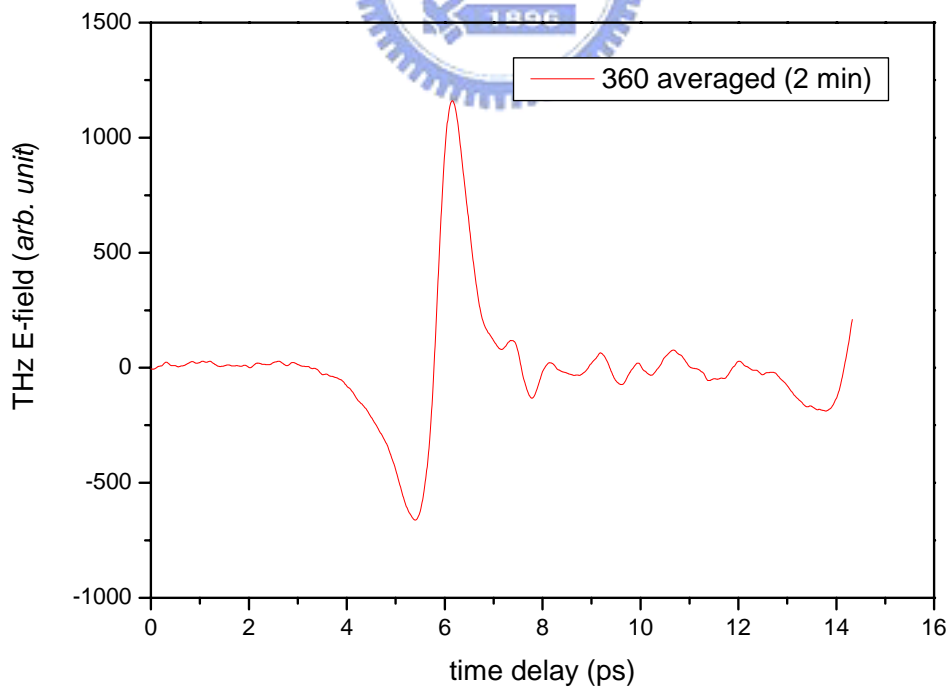


圖 4-6.d 360 筆 waveform

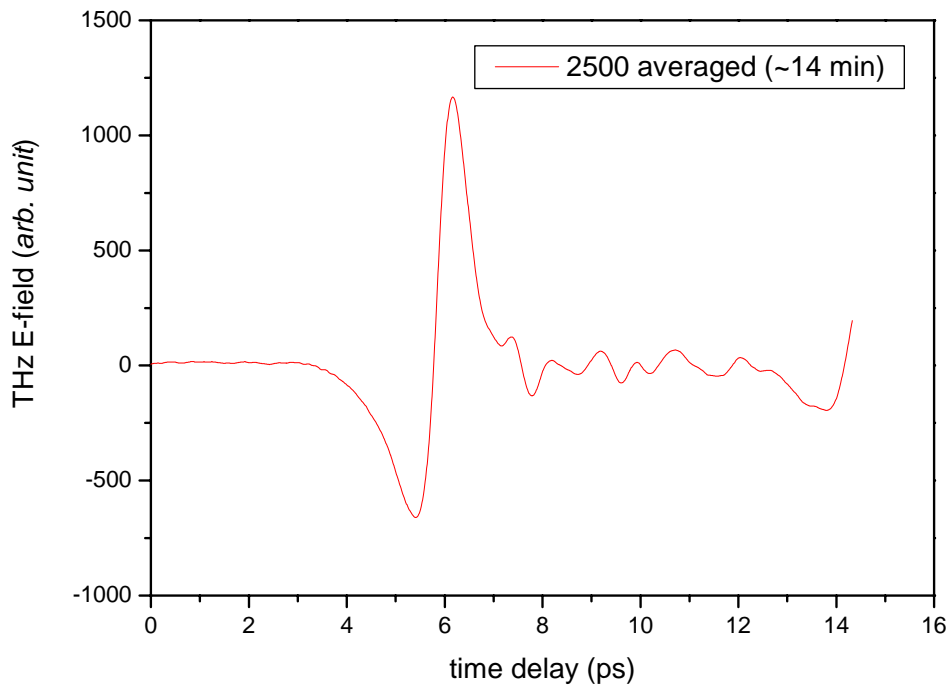


圖 4-6.e 2500 筆 waveform

平均筆數	所花時間(sec.)	SNR
1	0.33	10
10	3.33	41
100	33.3	131
180	60	134
360	120	129
500	166	212
1000	333.3	286
2500	833.33	436

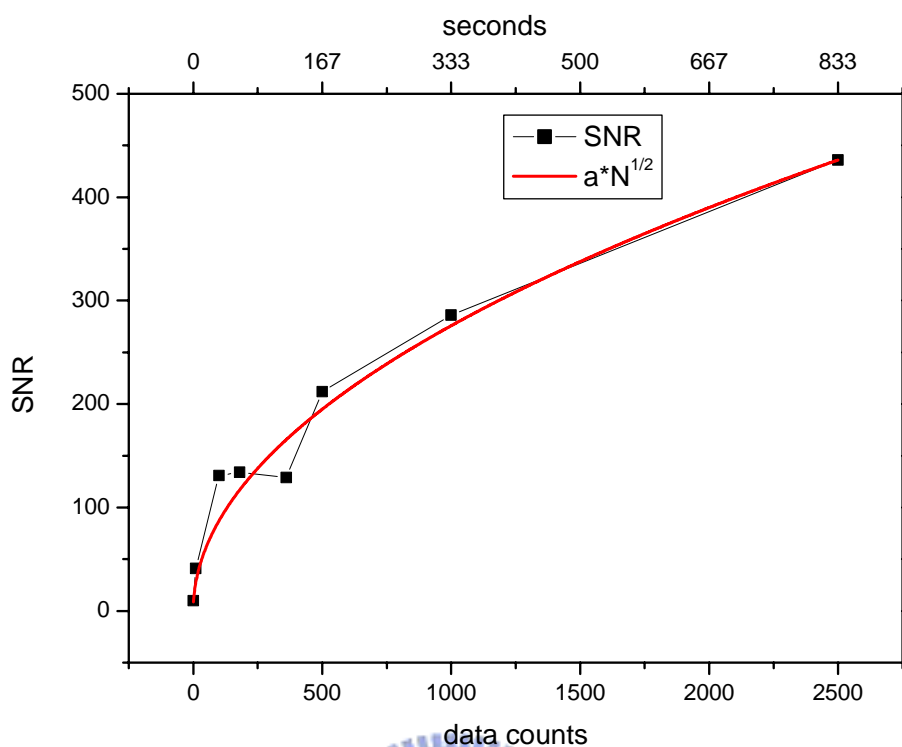


圖 4-6.f 訊噪比與平均筆數關係以及與 $N^{1/2}$ 之比較

理論上，如果平均次數增加 N 倍，則雜訊應該要下降 $N^{1/2}$ 倍，也就是訊噪比應該要上升 $N^{1/2}$ 倍，我們也將所得結果與 $N^{1/2}$ 作了一個比較，從圖中可以看到除了少數點以外，絕大多數區域都符合這 $N^{1/2}$ 的趨勢。

4.7 以快速掃描作 THz 成像

我們將已架好的快速掃描系統應用在 THz 成像上，使用這樣的架構，我們可以將點數取得足夠密。我們這次所掃的樣品仍然是火柴

盒，每個像素的 THz waveform 取 1024 點，鎖相放大器所使用的時間常數為 3ms，光學延遲線的掃描頻率為 1Hz，每個 THz waveform 平均筆數為 60 筆，每個 waveform 所花時間為一分鐘。兩次掃描所使用之參數比較如下：



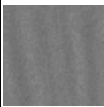
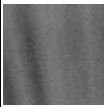
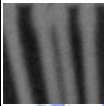
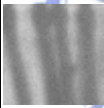
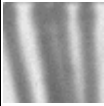
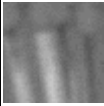
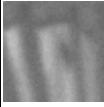
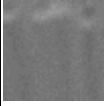
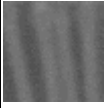
	步進馬達	快速掃描
掃描尺寸	38 × 50	50 × 50
掃描大小	1.9 cm × 2.5cm	2.5cm × 2.5cm
每像素間距	500 nm	400 nm
所花時間	65 hours	46 hours
每 waveform 掃描點數	256	1024

由上面表格可以看到已快速掃描所掃的尺寸為 50 × 50，總共 2500 個像素，所花的時間為 46 小時；而以步進馬達所掃的尺寸為 38 × 50，總共 1900 個像素，所花的時間卻要 65 小時。若我們將條件統一，讓步進馬達也掃 50 × 50 × 1024 的話，則所需的時間要 $65 \times (2500/1900) \times 4 = 342 \text{ hours} = \mathbf{14 \text{ days } 6 \text{ hours}}$ 。

我們可以看到以快速掃描的方式可以使我們以更少的時間來取得更好的 THz waveform。

所掃得到的 THz 成像圖分別如下：

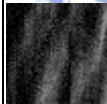
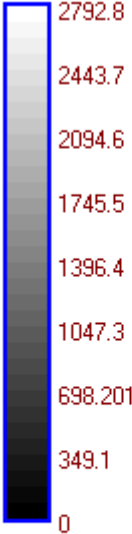
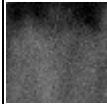
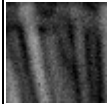
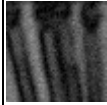
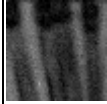
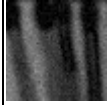
以不同時間延遲成像

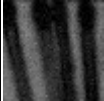
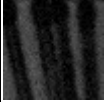
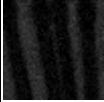
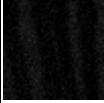

Time Delay (ps)	Image	
0		 <p>1789.3 1365.92 942.544 519.165 95.7864 -327.592 -750.971 -1174.35 -1597.73</p>
9.97		
11.33		
12.05		
12.51		
12.82		
14.51		
15.20		
17.84		
20.38		

23.20		
-------	---	--

在不同時間延遲所成的像都不同，我們可以看到在有些時間延遲上是黑火柴白背景，有些則相反。以時間延遲 15.2ps 為例，我們還可同時看到火柴頭與火柴棒的分別，在時間延遲 17.84 只看得到火柴頭，而在 23.20 卻只看到火柴棒。

以頻率振幅成像

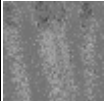
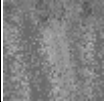
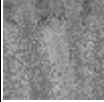
Frequency (THz)	Image	
0		
0.1		
0.2		
0.3		
0.4		
0.5		

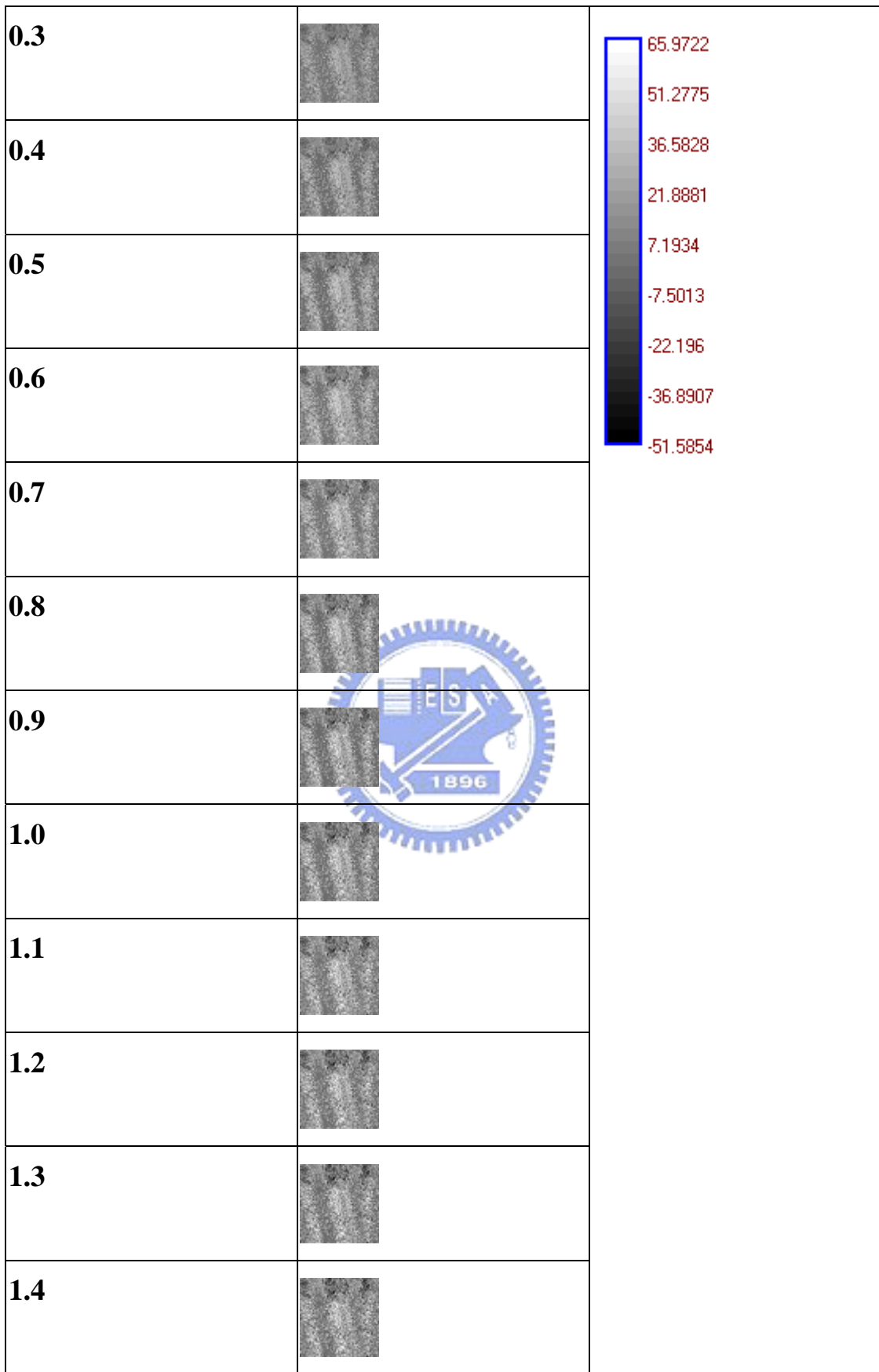
0.6		
0.7		
0.8		
0.9		
1.0		



以振幅來成像，可以看到在 0.1THz 處的影像非常模糊，而隨著頻率變高而更清楚。然而由於高頻振幅下降很多，以對比度亦下降至無法分辨。



以頻率相位成像

Frequency (THz)	Image	
0		
0.1		
0.2		



1.5		
1.95		

在頻域相位成像中，即使在高頻部分，我們仍然可以看到影像，而振幅成像在高頻對比度相當低無法清楚分辨。

4.8 像素 waveform 之討論

在不同位置上，我們所量到的 THz waveform 也會不同，在本小節中，我們將一些具有代表性點上的 waveform 取出來討論，使我們對其成像特性更了解。



如圖 4-7.a 所示，分別為火柴盒空白處、火柴棒、火柴頭的 THz waveform，我們可以看到空白處的 waveform 其峰值所在位置約為 12.5 ps，後面有一個小的信號是 THz 信號在天線中二次反射所造成的。火柴棒處所量到的 waveform，我們可以看到其峰值所在位置移動到 15.1 ps 處，由於散射、吸收等緣故其最大值也下降到 790 左右。火柴頭處所量到的 waveform，其峰值位置移至 18.9ps 處，信號強度下降更多。如果聚焦的焦點剛好打到一半火柴棒，一半空白處，我們便會看到這兩個 waveform 的干涉，如圖 4-7.b 所示，粗線為在邊界

時所量測到的 waveform，我們可以看到約有兩個峰值其位置與空白處與火柴棒處的位置相同。細線為將空白處與火柴棒的 waveform 各乘以一係數在相加的結果，其所得 waveform 與實際所量到的相當吻合。

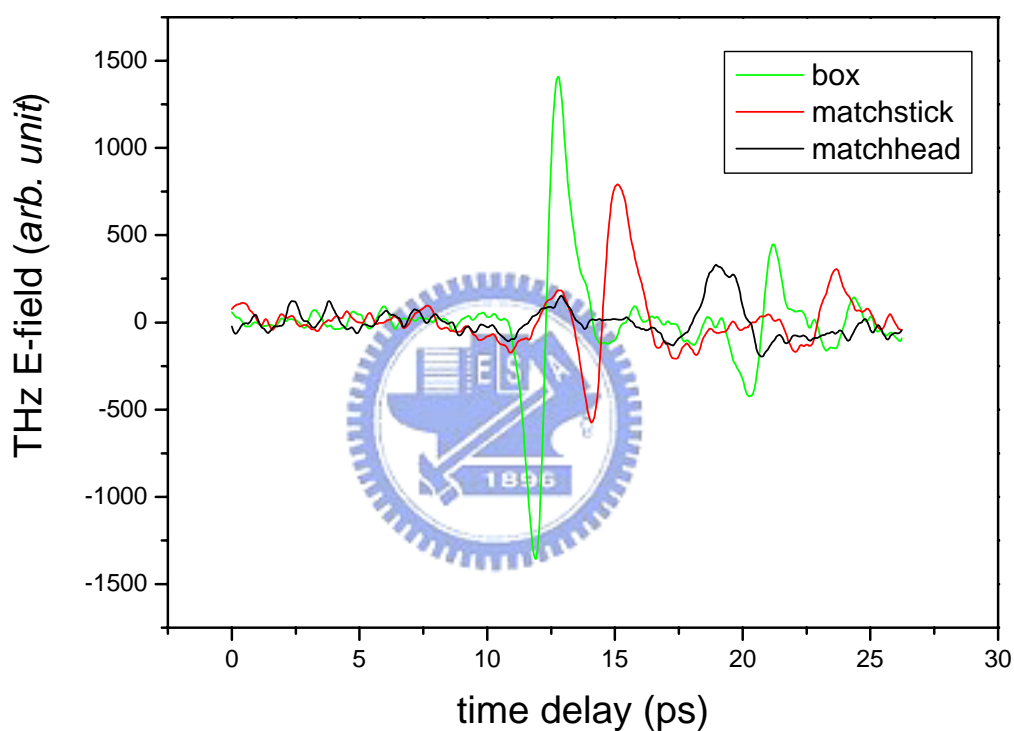


圖 4-7.a 火柴盒內各處的 THz waveform

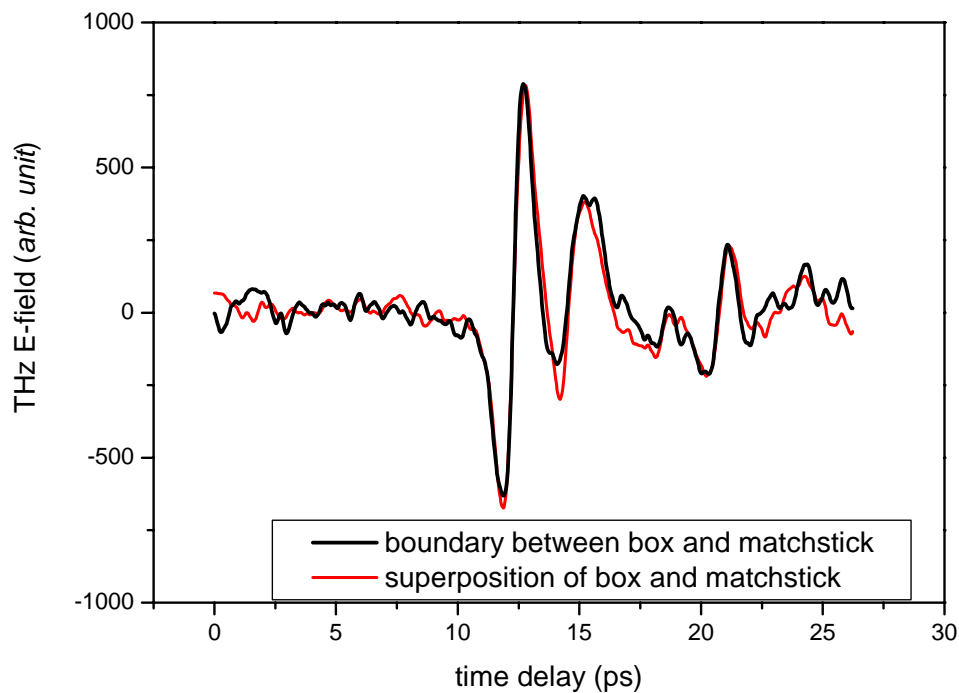


圖 4-7.b 空白-火柴棒處的 THz waveform 與空白及火柴棒相加之比較



另外一方面由於在天線內二次反射的關係，故在主要信號後面還有個小的信號，這小的信號會使得傅氏轉換之後的頻譜會呈鋸齒狀的分佈，如圖 4-8 所示，而沒有這反射小信號的傅氏轉換為一平滑之分佈如圖 4-3.c 所示。這鋸齒狀的頻譜分佈會造成在作頻域振幅成像時會有一亮一暗的現象發生。假設沒有後面反射信號所得的傅氏轉換為 $A(\omega)$ ，則後面的反射信號的傅氏轉換可以表示成 $a \times A(\omega)e^{i\Delta\theta(\omega)}$ 。 a 是代表反射的強度較弱，而 $e^{i\Delta\theta(\omega)}$ 則是由於時間上有位移則在傅氏轉換上會造成相延遲。這兩個相加的傅氏轉換可表示成

$$\begin{aligned}
& A(\omega) + aA(\omega)e^{i\Delta\theta(\omega)} \\
&= A(\omega)(1 + ae^{i\Delta\theta(\omega)}) \\
&= A(\omega)a'(\omega)e^{i\Delta\theta'(\omega)}
\end{aligned}$$

$ae^{i\Delta\theta(\omega)}$ 在極座標上是一個半徑為 a 的圓，它會與 1 隨著不同 $\Delta\theta(\omega)$ 相加減而使 $a'(\omega)$ 有大小的變化。如圖 4-9 所示為假設 $a=0.3$ 所作的 $a'(\omega)$ 對不同 $\Delta\theta$ 的關係，我們可以看到其呈現一個弦波的分佈，如果我們將圖 4-3.c 的頻譜分佈乘以這個弦波便會得到如圖 4-8 的分佈。

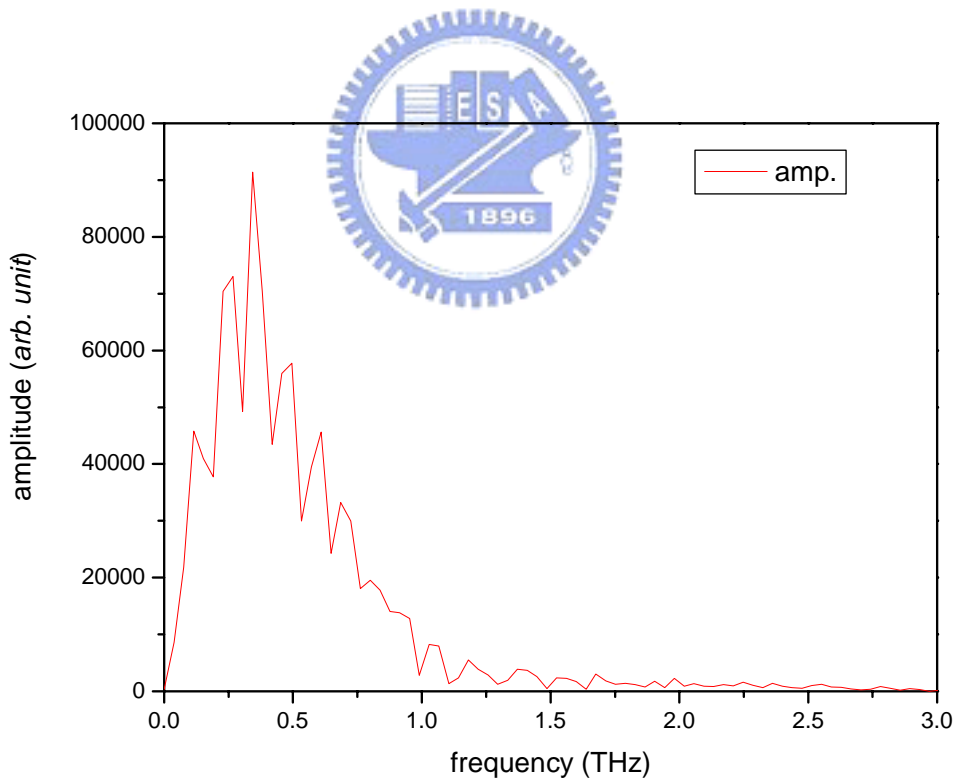


圖 4-8 鋸齒狀的頻譜分佈

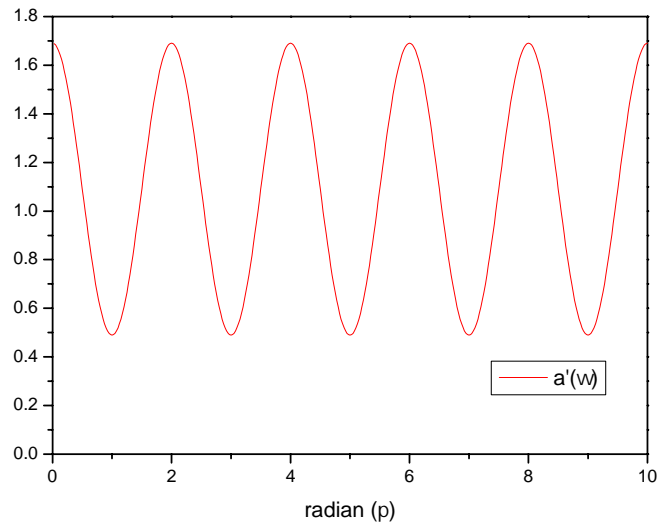


圖 4-9 $a'(\omega)$ 對不同 $\Delta\theta$ 的關係



第五章 THz Imaging 解析度

5.1 THz Imaging 解析度的限制

THz 成像的解析度主要是限制於其波長，根據基本光學理論，一道光束所能聚焦的最小大小約為其波長的尺寸。我們以光導天線所發出來的 THz 信號其頻率分布約從 0 ~ 1THz，中心頻率約在 0.3THz，我們將 1THz 換算成波長可以得到約 300 μ m 左右的大小，我們的解析度的限制大約也是在這個範圍。提高解析度的方式有幾個，我們可以提高 THz 的頻寬，或在待測物前面放一個孔徑(aperture)用來限制光點的大小，將待測物貼在發射器後或者是偵測器前也可以有效果。 [9]

5.2 本系統之解析度測試

我們製作了一個樣品，在 80 μ m 厚的不銹鋼片上刻了 LEO 等字，其中 L 的線寬為 400 μ m，E 的線寬為 500 μ m，O 的線寬為 600 μ m。然而在實際刻出來的寬度 L 約為 380 μ m，E 約為 480 μ m，O 約為 580 μ m



圖 5-1 樣品實際照片

。我們首先以各像素間距 200 μ m 來掃 LEO 這三個字，其結果如圖 5-2 所示，這張圖的掃描尺寸為 80 \times 35，掃描大小為 16mm \times 7mm。我們可以看到這張圖中似乎直線都看起來較粗，橫線看起來較細，如 L 字的直線線寬幾乎快把其橫線給蓋住了。我們分析這張圖，發現 LEO 這幾個字直線的線寬幾乎都是 1.4mm，不論其原來線寬是多少，然而字顏色的深度卻不同，L 的顏色最淺，O 的顏色最深。再看橫線部分其線寬大約是 0.8mm，這個不對稱的線寬我們認為是由於我們使用的天線 gap 較大所造成的，當 pump beam 打在 2mm gap 上面，然而實際發出 THz 信號的區域並不會是整個光點，而是實際光電流的導電區域，而這應該是一個橫跨這 2mm gap 的直線，如此便造成了這個上下左右不對稱的線寬。

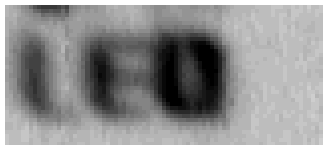
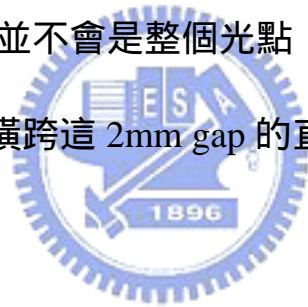


圖 5-2 以每點間距 200 μ m 來掃 LEO 三個字，其中 L 線寬為 400 μ m，E 為 500 μ m，O 為 600 μ m

我們想要知道掃描時每點像素的間距是否對於解析度有影響，便將同樣的樣品改以 100 μ m 來掃如圖 5-3 所示。這張圖的尺寸為 120 \times 70，掃描大小為 12mm \times 7mm。我們可以看到上下左右不對稱的

線寬依舊存在。而解析度方面似乎只有改善幅度不大，以 O 字來看，其線寬便得似乎較為細。其直線線寬(垂直方向)寬度約為 1.3mm，橫線(水平方向)線寬約 0.8mm，與原來相去不大。由以上結果看來，將掃描的間距變密只能提高邊界的解析度，與整體的解析度的幫助其實並不很大。

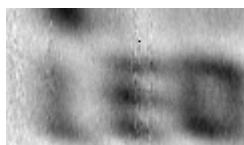


圖 5-3 以每點間距 100mm 來掃 LEO 三個字,其中 L 線寬為 400mm,E 為 500mm, O 為 600mm



為了要知道我們系統 THz 在焦點上的 spotsize 大小，我們使用刀口法[12]來量取焦點上的 spotsize，其結果如圖 5-4.a、圖 5-4.b 所示，我們分別對水平方向(X)以及垂直方向(Y)來掃。我們計算寬度的方式為先將所得結果平方，將電場對空間分佈轉成光強度對空間分佈。再計算其強度 90% 與 10% 處的寬度，再將所得寬度乘以 0.78 圖 5-3.a 為水平方向所掃的結果，計算出 spotsize 大小約為 1.64mm。垂直方向如圖 5-3.b 所示，計算得 spotsize 約為 0.69mm。如此我們可以看到 spotsize 尺寸約為 1.64mm×0.69mm，長寬比為 2.37。

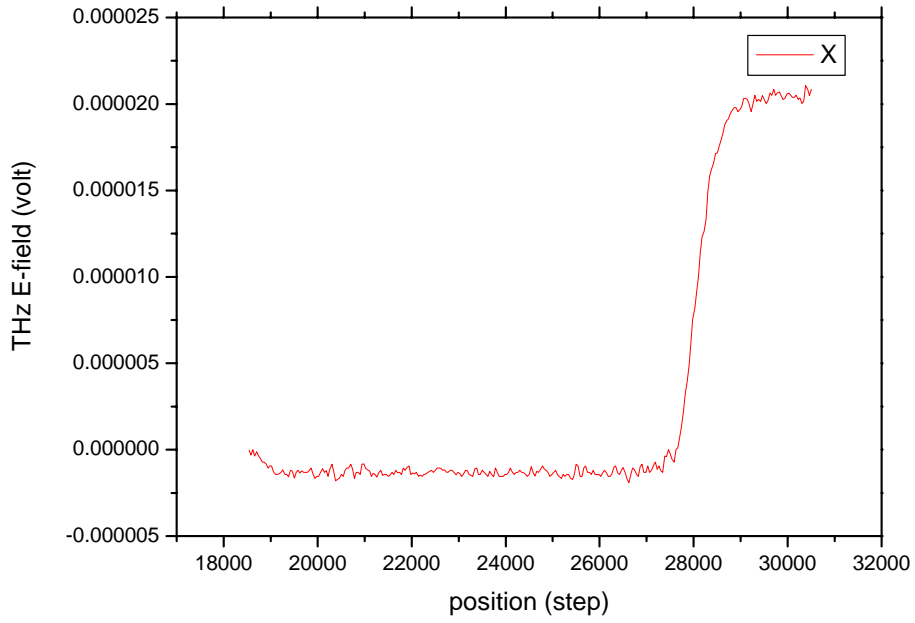


圖 5-4.a 以刀口法量取水平方向之 spotsizes

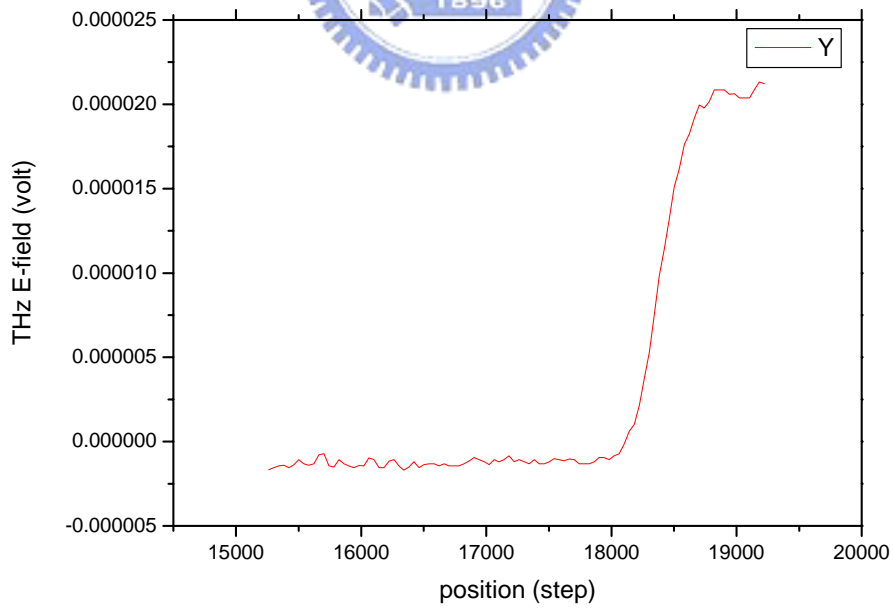


圖 5-4.b 以刀口法量取垂直方向之 spotsizes

5.3 以小 gap 天線來作成像

在上節中我們可以看到我們成像的水平垂直方向寬度的不對稱，我們希望可以改善這缺點，便換了一個小 gap 的天線來作實驗。這天線的尺寸兩條金線寬度是 10mm，gap 寬度為 5mm。由於此天線之 gap 比所發出之 THz 信號之波長還小很多，故其不對稱性是可被忽略的。我們將所量到的 5mm gap 信號與同期所量到的 1.5mm gap 的信號比較如圖 5-5 所示，我們可看到以小 gap 所量到的信號之 peak 值強度約為 20×10^{-6} 而以大 gap 所量到的約為 70×10^{-6} ，振幅強度約為 1/3 至 1/4。

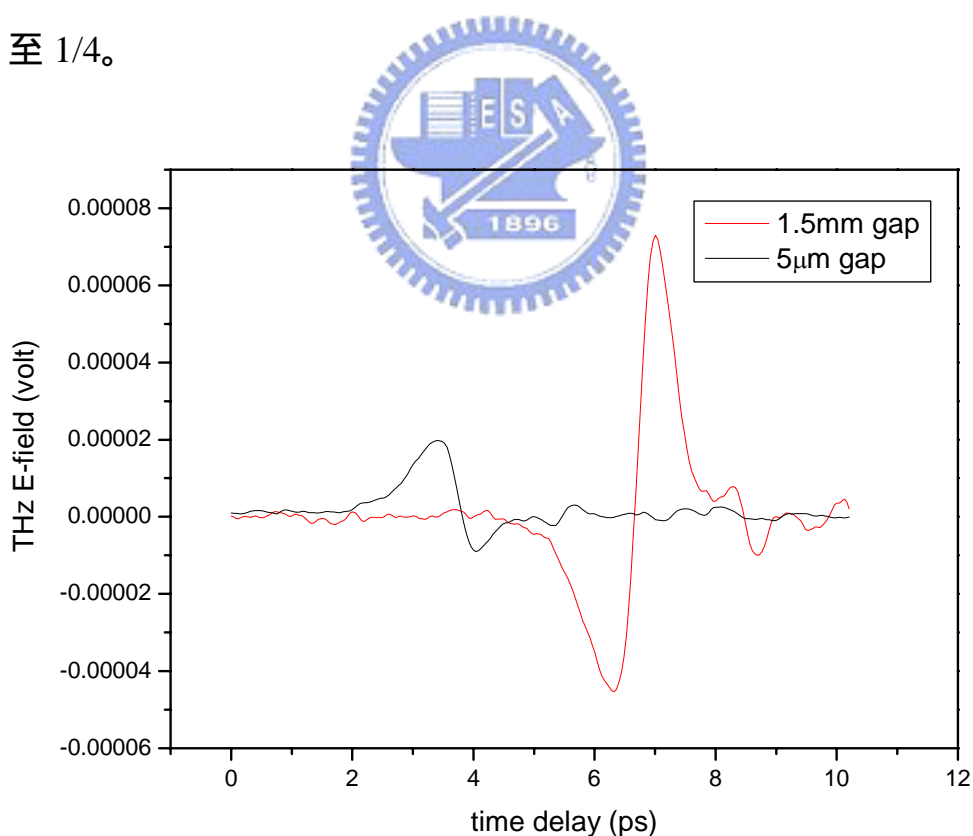


圖 5-5 以 5mm gap 所量到的 THz 信號與 10mm gap 的信號之比較

我們以此天線來作 THz 成像其結果如圖 5-6 所示，我們可以看到解析度有了不錯的提昇。其直線(垂直)線寬約為 1.0mm，橫線(水平)方向線寬約 0.8mm，雖然似乎還是有些不對稱，但我們可以看到改以小 gap 天線之後，不論是解析度或者是垂直水平方向之對稱性都有了不錯的改善。

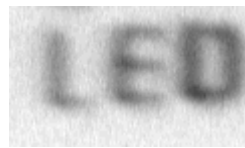


圖 5-6 以 5mm gap 所量到之 THz Imaging

為了要對這小 gap 的解析度有更深的了解，我們同樣以刀口法來量其 spotsize，其結果如圖 5-7.a、圖 5-7.b。在水平方向，所量到的 spotsize 約為 1.58mm 左右。而在垂直方向，所量到 spotsize 約為 1.17mm，長寬比約為 1.35。雖然 spotsize 的大小並未有明顯的改善，但我們所量到的影像解析度卻有不錯的改善。

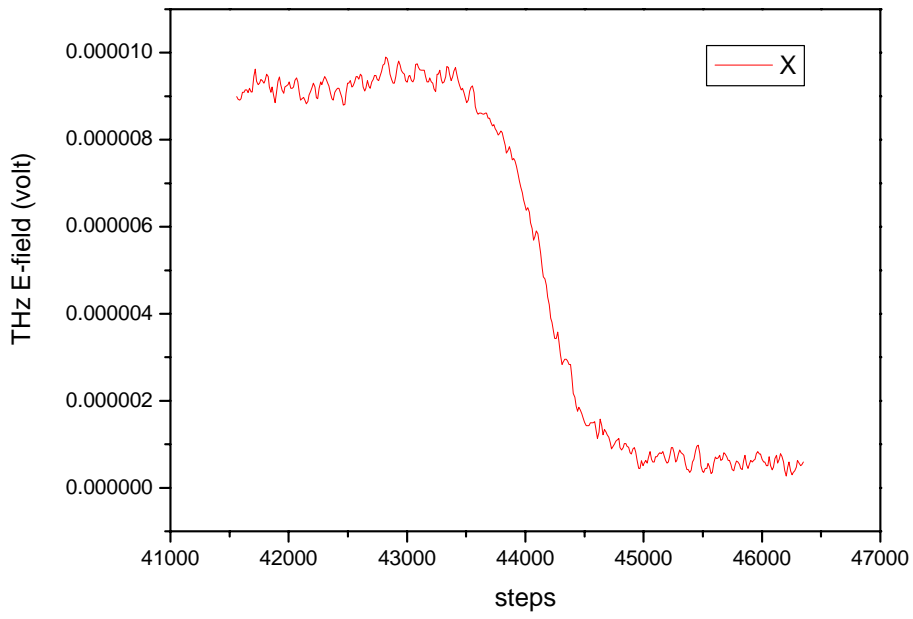


圖 5-7.a 以刀口法量取水平方向之 spotsize

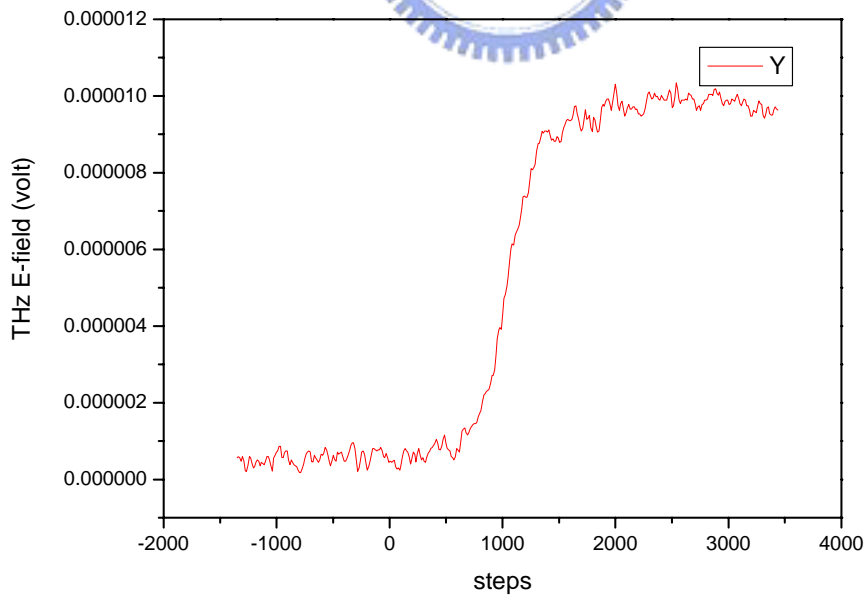


圖 5-7.b 以刀口法量取垂直方向之 spotsize

第六章 結論與未來展望

我們架設了一個具有快速掃描功能的 THz Imaging 系統，並用其對火柴盒做掃描，證實了 THz 的高穿透性。本系統以快速掃描方式所得到的訊噪比能以 33 秒鐘達到 131，兩分鐘半達到 212，十四分鐘達到 436。以上是電場的訊噪比，換算成光強度則要做一次平方。我們也初步對系統的解析度做分析，並改用小 gap 之天線來做成像，其解析度約可達到 $400\mu\text{m}$ 。

未來我們希望可以將鎖相放大器拿掉來做快速掃描，如此可以得到快速掃描的全速。另一方面，我們也希望可以近場光學等方式來提昇更高系統之解析度。



附錄 A :

在本附錄中收錄了用來作快速掃描以及 THz Imaging 的程式碼。

Unit1.h

```
//-----  
  
#ifndef Unit1H  
#define Unit1H  
//-----  
  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
#include "MULTIPLELib_OCX.h"  
#include <Buttons.hpp>  
#include <ExtCtrls.hpp>  
#include <OleCtrls.hpp>  
#include <Dialogs.hpp>  
//-----  
  
class TForm1 : public TForm  
{  
    __published:    // IDE-managed Components  
        TBitBtn *BitBtn1;  
        TBitBtn *BitBtn2;  
        TImage *Image1;  
        TPci9812 *Pci98121;  
        TEdit *Edit1;  
        TEdit *Edit2;  
        TLabel *Label1;  
        TLabel *Label2;  
        TOpenDialog *OpenDialog1;  
        TPanel *Panel1;  
        TButton *Button1;
```



```

TEdit *Edit3;
TEdit *Edit4;
TEdit *Edit5;
TLabel *Label3;
TLabel *Label4;
TLabel *Label5;
TEdit *Edit6;
TLabel *Label6;
TLabel *Label7;
TPanel *Panel2;
TButton *Button2;
TEdit *Edit7;
TLabel *Label8;
TEdit *Edit8;
TLabel *Label9;
TEdit *Edit9;
TLabel *Label10;
TEdit *Edit10;
TLabel *Label11;
TCheckBox *CheckBox1;
TEdit *Edit11;
TLabel *Label12;
TEdit *Edit12;
TEdit *Edit13;
TLabel *Label13;
TLabel *Label14;
TButton *Button3;
TLabel *Label15;
TTimer *Timer1;
TLabel *Label16;
TLabel *Label17;
TLabel *Label18;
TButton *Button4;
TEdit *Edit14;
TLabel *Label19;

void __fastcall FormCreate(TObject *Sender);
void __fastcall BitBtn1Click(TObject *Sender);
void __fastcall Pci98121AiComplete(TObject *Sender,

```



```

    TVariant *ScaledData, TVariant *BinaryCodes);
void __fastcall Button1Click(TObject *Sender);
void __fastcall Button2Click(TObject *Sender);
void __fastcall Button3Click(TObject *Sender);
void __fastcall BitBtn2Click(TObject *Sender);
void __fastcall Timer1Timer(TObject *Sender);
void __fastcall Button4Click(TObject *Sender);

private:    // User declarations
public:
    int TimerCountDown;    //剩下時間
    bool ScanRunning;    //正在掃描中??
    TObject *Form1Sender;    // User declarations
    __fastcall TForm1(TComponent* Owner);
    void __fastcall averagedata(int Ndata, int * ch1, int * ch2, int blocksize,int NeededSize,float
*returndata,int averagecount);//平均 data
    int FindMin(int * data, int Tsize, int Sindex, int blocksize);//尋找最小值之 index
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif

```

Unit1.cpp

```

//-----

#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "Unit2.h"

```

```

#include <stdio.h>
//-----
#pragma package(smart_init)
#pragma link "MULTIPLELib_OCX"
#pragma resource "*.dfm"
TForm1 *Form1;
int Total_Run = 0;

//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----

void __fastcall TForm1::FormCreate(TObject *Sender)
{
    Pci98121->AIRange      = 0 ;
    Pci98121->CardType     = 4 ;
    Pci98121->channel      = 1 ;
    Pci98121->ClockFreq    = 0 ;
    Pci98121->ClockSource = 0 ;
    Pci98121->DoubleBufferMode = false ;

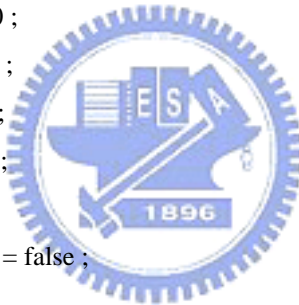
    Pci98121->OpenMode     = 0;
    Pci98121->ReturnType  = 0 ;

    Pci98121->StreamToFile = true ;
    Pci98121->TriggerMode  = 0 ;
    Pci98121->TriggerPolarity = 1 ;
    Pci98121->TriggerSource = 0 ;

    Pci98121->DeviceName  = "Damage" ;

    Edit1->Text = "data";
    Edit2->Text = "out.txt";
    Edit3->Text = "32000";
    Edit4->Text = "900";
    Edit5->Text = "1900";
}

```



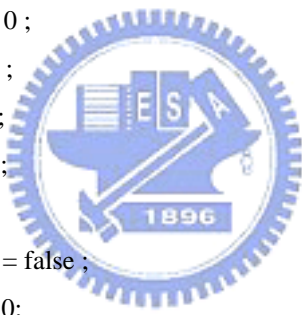
```

Edit6->Text = "9999";
Edit7->Text = "1900";
Edit8->Text = "190";
Edit9->Text = "aved2.txt";
Edit10->Text = "out\\aved.txt";
Edit11->Text = "1";
Edit12->Text = "4000";
Edit13->Text = "1000";
Edit14->Text = "1";

TimerCountDown = 0;
Label17->Caption = 0;
ScanRunning = false;
}
//-----
void __fastcall TForm1::Button3Click(TObject *Sender)
{
    Pci98121->AIRange      = 0;
    Pci98121->CardType     = 4;
    Pci98121->channel      = 1;
    Pci98121->ClockFreq    = 0;
    Pci98121->ClockSource  = 0;
    Pci98121->DoubleBufferMode = false;
    Pci98121->OpenMode     = 0;
    Pci98121->ReturnType  = 0;
    Pci98121->StreamToFile = true;
    Pci98121->TriggerMode  = 0;
    Pci98121->TriggerPolarity = 1;
    Pci98121->TriggerSource = 0;
}
//-----

void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{
    ScanRunning = true;
    Pci98121->NumOfScan   = StrToInt(Edit12->Text); //data 筆數
    Pci98121->ScanRate    = StrToInt(Edit13->Text); //掃描速度
    Pci98121->StartContAI("out\\"+Edit1->Text);
    Label15->Caption = "running...";
}

```




```

TimerCountDown = (int)(StrToInt(Edit12->Text)/StrToInt(Edit13->Text))+1;
Timer1->Enabled = true;

}
//-----
void __fastcall TForm1::Pci98121AiComplete(TObject *Sender,
    TVariant *ScaledData, TVariant *BinaryCodes)
{ //量測完後的處理函式，自動被呼叫
    Pci98121->StopContAI();
    Timer1->Enabled = false;
//    char *inFN=new char[256],outc1FN[256],outc2FN[256];
    FILE *sp1 , *in, *sp2 , *out , *sp3;
    int readin,n,readdata1,readdata2;

    Label15->Caption = "finished...";

    in = fopen(("out\\"+Edit1->Text).c_str(),"rb");
    out = fopen(("out\\"+Edit2->Text).c_str(), "w");

    fseek(in,60,SEEK_SET); //跳過檔頭資訊
    while(!feof(in)) //將 binary data 轉為 ASCII
    {
        fread(&readin,2,1,in);
        readdata1=(unsigned short)readin>>4;
        if(readdata1>=2048)
            readdata1 -= 4096;
        fread(&readin,2,1,in);
        readdata2=(unsigned short)readin>>4;
        if(readdata2>=2048)
            readdata2 -= 4096;
        fprintf(out,"%d  %d\n",readdata1,readdata2);
    };
    fclose(in);
    fclose(out);
    Button1Click(Sender);
/*
    if(Total_Run < StrToInt(Edit11->Text) )
        BitBtn1Click(Sender);

```

```

*/
    ScanRunning = false;
};

//=====================================================
void __fastcall TForm1::averagedata(int Ndata, int * ch1, int * ch2, int blocksize,int NeededSize,float
*returndata, int averagecount)
{ //將所得數據平均用副程式 , ch1 為 ODL 位置函數 , ch2 為 THz 信號
    int a,b,NumOfAve=0;
    long RetMinIndex,MinIndexNow=0;

    for(b=0;b<NeededSize;b++)
        *(returndata+b)=0;

    for(a=0;a<Ndata;a+=blocksize)
    {
        RetMinIndex=FindMin(ch1,Ndata,a,blocksize);
        if( (*(ch1+RetMinIndex)+5<*(ch1+a)) && (*(ch1+RetMinIndex)+5<*(ch1+a+blocksize)) )
            //此最小值比區間左右兩端值都小則成立
        if(RetMinIndex+NeededSize < Ndata){
            for(b=0;b<NeededSize;b++) //將 data 存入 buffer
                *(returndata+b)+=*(ch2+RetMinIndex+b);
            NumOfAve++;
            if(NumOfAve == averagecount)
                break;
        };
    };

    if(NumOfAve != 0)
    { for(b=0;b<NeededSize;b++)
        *(returndata+b)/=NumOfAve;
        Label7->Caption = NumOfAve;
    }
    else
        Label7->Caption = "No signal found";
}

int TForm1::FindMin(int * data, int Tsize, int Sindex, int blocksize) //尋找最小值所在 index

```

```

{ long a;
  long IndexOfMin=Sindex,minV=*(data+Sindex);

  for(a=0;Sindex+a<Tsize,a<blocksize;a++)
  { if(*(data+Sindex+a)<=minV)
    { minV=*(data+Sindex+a);
      IndexOfMin=Sindex+a;
    }
  };

  return IndexOfMin;
}

void __fastcall TForm1::Button1Click(TObject *Sender)
{ int a,b;
  int *ch1 , *ch2 ;
  float *receive;
  ch1 = new int[StrToInt(Edit3->Text)+5];
  ch2 = new int[StrToInt(Edit3->Text)+5];
  receive = new float[StrToInt(Edit5->Text)+5];

  FILE *in,*out;
  in = fopen(("out\\"+Edit2->Text).c_str() , "rb");
  for(int n=0 ; n < StrToInt(Edit3->Text) ; n++)
    fscanf(in,"%d %d",&ch1[n],&ch2[n]);
  fclose(in);
  int ave_count = StrToInt(Edit6->Text) ;

  averagedata(StrToInt(Edit3->Text),ch1,ch2,StrToInt(Edit4->Text),StrToInt(Edit5->Text),receive,ave_c
  ount);

  for(a=0;a<Edit5->Text.ToInt();a++)//尋找 waveform 的最小值及最大值並記錄給作 Imaging 參考
  { if(receive[a]>Form2->TDS_MAX)
    Form2->TDS_MAX = receive[a];
    if(receive[a]<Form2->TDS_MIN)
    Form2->TDS_MIN = receive[a];
  }; // records MAX and MIN

  out=fopen((Edit10->Text).c_str(),"w");

```

```

for(int n=0 ; n < StrToInt(Edit5->Text); n++)
    fprintf(out,"%f %f \n",n*Edit14->Text.ToDouble(),receive[n]);

fclose(out);

if(CheckBox1->Checked == true){ //draw waveform
    Image1->Canvas->Brush->Color = 0x0000;
    Image1->Canvas->FillRect(Rect(0,0,905,300));
    Image1->Width = 905 +10 ;
    Image1->Canvas->Pen->Color = clRed ;
    Image1->Canvas->Pen->Width = 2;
    Image1->Canvas->MoveTo(9,125);
    Image1->Canvas->LineTo(905,125);

    Image1->Canvas->Pen->Width = 1;
    Image1->Canvas->Pen->Color = clBlue ;
    Image1->Canvas->MoveTo(10,125);

    Image1->Canvas->MoveTo(10,(int) 125 - 100*(receive[0]/2048));

    for(int i = 1 ; i < StrToInt(Edit5->Text) ; i++)
        Image1->Canvas->LineTo(10+i*905.0/Edit5->Text.ToDouble(), (int) 125 -
100*(receive[i]/2048));
    }
delete ch1;
delete ch2;
delete receive;
}
//-----

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    FILE *in,*out;
    float *Sdata , *Ddata;
    Sdata = new float[StrToInt(Edit7->Text)+2];
    Ddata = new float[StrToInt(Edit8->Text)+10];

    float t;

```

```

for( int a=0 ; a <= StrToInt(Edit8->Text);a++)
    Ddata[a]=0.0;

if(OpenDialog1->Execute()){
    in=fopen((char*)(OpenDialog1->FileName.c_str()),"r");
    for(int a=0;a < StrToInt(Edit7->Text) ; a++)
        fscanf(in,"%f",&Sdata[a]);
    fclose(in);

    int Samp_rate;
    Samp_rate = (int)( StrToInt(Edit7->Text) / StrToInt(Edit8->Text) );
    for(int a=0 ; a<StrToInt(Edit8->Text) ; a++)
    { t=0;
        for(int b=0 ; b<Samp_rate ; b++)
            t+=Sdata[a*Samp_rate+b];
        Ddata[a]=t/Samp_rate;
    };

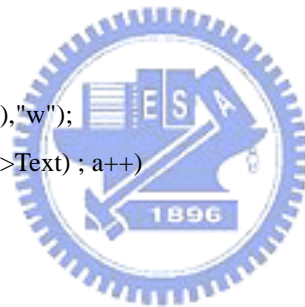
    out=fopen((Edit9->Text).c_str(),"w");
    for(int a=0 ; a< StrToInt(Edit8->Text) ; a++)
        fprintf(out,"%f\n",Ddata[a]);

    fclose(out);

};
}
//-----
void __fastcall TForm1::BitBtn2Click(TObject *Sender)//停止掃描
{ Pci98121->StopContAI();
    Label15->Caption = "stopped...";
    Timer1->Enabled = false;
    Label17->Caption = 0;

}
//-----
void __fastcall TForm1::Timer1Timer(TObject *Sender)//計算剩餘時間
{ TimerCountDown--;
    Label17->Caption = TimerCountDown;
}

```



```

}
//-----
void __fastcall TForm1::Button4Click(TObject *Sender)
{ Form2->Visible = true;

}
//-----

```

Unit2.h

```

//-----
#ifndef Unit2H
#define Unit2H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Buttons.hpp>
#include <ExtCtrls.hpp>
//-----
class TForm2 : public TForm
{
__published:    // IDE-managed Components
    TPanel *Panel1;
    TLabel *Label5;
    TPanel *Panel2;
    TLabel *Label6;
    TLabel *Label7;
    TLabel *Label8;
    TLabel *Label9;
    TEdit *Edit1;
    TEdit *Edit2;
    TEdit *Edit3;
    TEdit *Edit4;

```



```

TPanel *Panel3;
TLabel *Label10;
TEdit *Edit5;
TBitBtn *BitBtn1;
TPanel *Panel4;
TButton *Button1;
TButton *Button2;
TLabel *Label1;
TLabel *Label2;
TEdit *Edit6;
TLabel *Label3;
TEdit *Edit7;
TTimer *Timer1;
TLabel *Label4;
TLabel *Label11;
TLabel *Label12;
TLabel *Label13;
TLabel *Label14;
TTimer *Timer2;
TLabel *Label15;
void __fastcall BitBtn1Click(TObject *Sender);
void __fastcall Timer1Timer(TObject *Sender);
void __fastcall Timer2Timer(TObject *Sender);
void __fastcall Button1Click(TObject *Sender);
void __fastcall Button2Click(TObject *Sender);
private: // User declarations
public:
    int pstX; //2D 平移台現在 X 位置
    int pstY; //2D 平移台現在 Y 位置
    TObject * SenderOBJ;
    bool IsTmProcRun; //是否正在執行中??
    int timerN; //總共已掃像素
    long leftsecs;
    float TDS_MAX; //時域最大值(所有像素中)
    float TDS_MIN; //時域最小值(所有像素中)
    bool Form2ScanOver; // User declarations
    __fastcall TForm2(TComponent* Owner);
    MoveTo(int X, int Y);

```



```

};
//-----
extern PACKAGE TForm2 *Form2;
//-----
#endif

```

Unit2.cpp

```

//-----

#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
#include "Unit2.h"
#include<D:\eo90\wu\RS232 專區\RS232Comm.h>
#include<stdio.h>
#include<stdlib.h>
//-----

#pragma package(smart_init)
#pragma resource "*.dfm"
TForm2 *Form2;
RS232Comm Com;

//-----

__fastcall TForm2::TForm2(TComponent* Owner)
    : TForm(Owner)
{
    Com.RS232Init("COM1");
    IsTmProcRun = false;
}
//-----

void __fastcall TForm2::BitBtn1Click(TObject *Sender)//THz Imaging 開始
{ char temp[50],temp1[15],temp2[15];
  AnsiString readtemp;

```




```

int a,b;
FILE *out;

Label15->Caption = "Running...";

Com.Send(".,@0SX0,,,,",10);
Sleep(500);
Com.Send(".,@0SY0,,,,",10);
Sleep(500);

Panel2->Enabled = false;
pstX=0;
pstY=0;
timerN=-1;

out=fopen((Edit5->Text+"Image.inf").c_str(),"w"); //儲存基本資訊供 Imaging 使用
fprintf(out,"%d %d %d %d %f %d\n",Edit1->Text.ToInt(),Edit2->Text.ToInt(),
        Edit3->Text.ToInt(),Edit4->Text.ToInt(),
        Form1->Edit14->Text.ToDouble(),Form1->Edit5->Text.ToInt());
fclose(out);

TDS_MAX = 0;
TDS_MIN = 0;
Form2ScanOver = false;

Timer1->Enabled = true;
Timer2->Enabled = true;

}
//-----

TForm2::MoveTo(int X, int Y)//移動 2D 平移台
{
    AnsiString t;
    t=".,@0PX" + t.CurrToStr((X-pstX)*Edit3->Text.ToInt());
    t+= ".,";
    Com.Send(t.c_str(),t.Length());
    Sleep(500+ abs((X-pstX)*Edit3->Text.ToInt()*0.5);
    t=".,@0PY" + t.CurrToStr((Y-pstY)*Edit4->Text.ToInt());
}

```



```

t+= ",";
Com.Send(t.c_str(),t.Length());
Sleep(500+ abs((Y-pstY)*Edit3->Text.ToInt()*0.5);

pstX=X;
pstY=Y;
}
void __fastcall TForm2::Timer1Timer(TObject *Sender) //控制 waveform 之掃描
{ if(IsTmProcRun)
return;
if(Form1->ScanRunning)
return;

IsTmProcRun = true;

int a,b;
timerN++;
b= timerN % Edit2->Text.ToInt(); //計算 Y 軸座標
a= (timerN-b)/Edit1->Text.ToInt(); //計算 X 軸座標

MoveTo(a,b);
Label11->Caption = a;
Label12->Caption = b;
leftsecs =
(Edit1->Text.ToInt()*Edit2->Text.ToInt()-timerN)*(Form1->Edit12->Text.ToInt()/Form1->Edit13->Text.ToInt());
Edit7->Text = (timerN+1) % Edit2->Text.ToInt();
Edit6->Text = (timerN+1-Edit7->Text.ToInt())/Edit1->Text.ToInt();

Form1->Edit10->Text = Form2->Edit5->Text + a + " " + b + ".txt";
Form1->BitBtn1Click(Sender);

if(timerN >= (Edit1->Text.ToInt()*(Edit2->Text.ToInt()-1))
{ Form2ScanOver = true;
Timer1->Enabled = false;
};
IsTmProcRun = false;
}

```



```

//-----

void __fastcall TForm2::Timer2Timer(TObject *Sender)//計算剩餘時間
{ int h,m,s;
  AnsiString t;
  leftsecs--;
  s=leftsecs%60;
  m=(leftsecs-s)/60%60;
  h=(leftsecs-s)/3600;
  Label14->Caption = t.CurrToStr(h) + "  hours  " + t.CurrToStr(m) + "  minutes  " +
t.CurrToStr(s) + "  seconds ";

  if(Form2ScanOver == true && Form1->ScanRunning == false)
  { Timer2->Enabled = false;
    Label15->Caption = "finished...";

    FILE *out =fopen((Edit5->Text+"Image.inf").c_str(),"a");
    fprintf(out,"%f %f \ncompleted",TDS_MAX,TDS_MIN);
    fclose(out);
  };
}
//-----

void __fastcall TForm2::Button1Click(TObject *Sender)//中斷功能
{ Timer1->Enabled = false;
  Timer2->Enabled = false;
  Label15->Caption = "wait form1 to finish...";
}
//-----

void __fastcall TForm2::Button2Click(TObject *Sender)//恢復掃描功能
{
  timerN=Edit6->Text.ToInt()*Edit2->Text.ToInt()+Edit7->Text.ToInt()-1;
  Timer1->Enabled = true;
  Timer2->Enabled = true;
}
//-----

```



附錄 B :

在此收錄將所得數據成像的程式。

Unit1.h

```
//-----  
  
#ifndef Unit1H  
#define Unit1H  
//-----  
#include <Classes.hpp>  
#include <Controls.hpp>  
#include <StdCtrls.hpp>  
#include <Forms.hpp>  
#include <Dialogs.hpp>  
#include <ExtCtrls.hpp>  
#include <Menus.hpp>  
//-----  
class TMainForm : public TForm  
{  
    __published:    // IDE-managed Components  
        TMainMenu *MainMenu1;  
        TMenuItem *FILE1;  
        TMenuItem *LoadTimeDomainData1;  
        TMenuItem *LoadFrequencyDomainData1;  
        TMenuItem *SaveImage1;  
        TMenuItem *FourierTransform1;  
        TMenuItem *FOURIERTRANSFORM2;  
        TMenuItem *N1;  
        TMenuItem *Setup1;  
        TMenuItem *MImage1;  
        TMenuItem *Mimage2;  
        TMenuItem *FrequencyDomain1;  
        TMenuItem *FrequencyDomainphase1;
```



```

TMenuItem *N2;
TMenuItem *N3;
TMenuItem *ExportIntensityPIC1;
TOpenDialog *OpenDialog1;
TSaveDialog *SaveDialog1;
TOpenDialog *OpenDialog2;
TLabel *Label1;
TLabel *Label2;

TImage *Image1;
TImage *Image3;
TMenuItem *TimeDomainImages1;
TMenuItem *FrequencyDomainImages1;
void __fastcall LoadTimeDomainData1Click(TObject *Sender);
void __fastcall SaveImage1Click(TObject *Sender);
void __fastcall Mimage2Click(TObject *Sender);
void __fastcall FOURIERTRANSFORM2Click(TObject *Sender);
void __fastcall TimeDomain1Click(TObject *Sender);
void __fastcall FrequencyDomain1Click(TObject *Sender);
void __fastcall FrequencyDomainphase1Click(TObject *Sender);
void __fastcall ExportIntensityPIC1Click(TObject *Sender);
void __fastcall Setup1Click(TObject *Sender);
void __fastcall LoadFrequencyDomainData1Click(TObject *Sender);
void __fastcall N2Click(TObject *Sender);
private: // User declarations
public:
    AnsiString InfoFile; //資訊檔
    AnsiString BaseDirectory; //Imaging 數據存放路徑
    int Xpoints; //X 方向總共點數
    int Ypoints; //Y 方向總共點數
    int Xres; //X 方向每像素間距
    int Yres; //Y 方向每像素間距
    AnsiString FourierBaseDirectory; //傅氏轉換後數據存放路徑
    AnsiString ImageBaseDirectory; //時域成像存放路徑
    AnsiString FourierImageABaseDirectory; //頻域振幅成像存放路徑
    AnsiString FourierImagePBaseDirectory; //頻域相位成像存放路徑
    float deltaT; //每點差多少時間延遲
    int TDpoints; //每個 waveform 有多少點

```

```

float TDMAX; //TD:時域 FD:頻域
float TDMIN; // MAX:最大值 MIN:最小值
float FDAMAX; //A:amplitude P:phase
float FDAMIN;
float FDPMAX;
float FDPMIN;
float FDSfreq; //頻域傅氏轉換起始頻率
float FDEfreq; //頻域傅氏轉換截止頻率
float FDdf;

TComponent *AOwner;

// User declarations
__fastcall TMainForm(TComponent* Owner);
protected:
FT(char *inPath,char *outPath,double dt,double SF,double EF,double df);
IntensityPic(Double Max, Double Min);

};
//-----
extern PACKAGE TMainForm *MainForm;
//-----
#endif

```



Unit1.cpp

```

//-----

#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "Unit2.h"
//-----

#pragma package(smart_init)
#pragma resource "*.dfm"

#include<math.h>

```

```

#include<complex.h>

TMainForm *MainForm;

//-----
__fastcall TMainForm::TMainForm(TComponent* Owner)
    : TForm(Owner)
{
    FDSfreq=-2;
    FDEfreq=4;
    FDdf=0.05;
}
//-----

void __fastcall TMainForm::LoadTimeDomainData1Click(TObject *Sender)//載入時域數據的資訊檔
{
    FILE *inf_file;
    if(OpenDialog1->Execute())
    {
        InfoFile=OpenDialog1->FileName;
        BaseDirectory=InfoFile.SubString(1,InfoFile.LastDelimiter("\\"));
        Label1->Caption=InfoFile.SubString(1,InfoFile.LastDelimiter("\\"));
        inf_file=fopen(InfoFile.c_str(),"r");
        fscanf(inf_file,"%d %d %d %d %f %d %f %f",&Xpoints,&Ypoints,
            &Xres,&Yres,&deltaT,&TDpoints,&TDMAX,&TDMIN);
        fclose(inf_file);
        TDMAX*=1.2;
        TDMIN*=1.2;

        FourierTransform1->Enabled=true; //開啟傅式轉換功能
        TimeDomainImages1->Enabled=true; //開啟時域數據相關功能
        FrequencyDomainImages1->Enabled=false; //關閉頻域相關功能
    }
};

}
//-----

void __fastcall TMainForm::SaveImage1Click(TObject *Sender)
{
    AnsiString temp;

```

```

if(SaveDialog1->Execute())
{ temp=SaveDialog1->FileName + ".bmp" ;
  Image3->Picture->SaveToFile(temp);
  Label2->Caption=temp;
};

}

//-----
void __fastcall TMainForm::Mimage2Click(TObject *Sender)//固定時間延遲成像用
{ int a,b;
  int x,y;
  float T;
  FILE *in;
  Image1->Width=120;
  Image1->Height=70;

  in=fopen("E:\\THz Imaging\\0716\\LEO.txt","r");
  for(a=0;a<=2800;a++)
  { fscanf(in,"%d %d %f",&x,&y,&T);

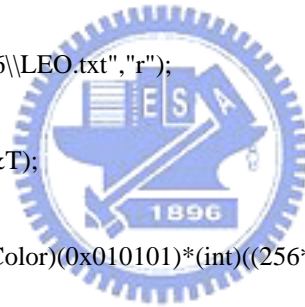
Image1->Canvas->Pixels[x][y]=(TColor)(0x010101)*(int)((256*(T-(-0.5)*0.000001)/(15*0.000001)));
};
  Image1->Picture->SaveToFile("E:\\THz Imaging\\0716\\LEO.bmp");
  fclose(in);

}

//-----

TMainForm::FT(char *inPath,char *outPath,double dt,double SF,double EF, double df)
//fourier transform 函式
{ double *amp;
  double *t;
  FILE *infile;
  FILE *outfile;
  int points,BasePhase=0;
  double a,b,c,d,temp;
// double dt;

```




```

double freq;
complex<double> re,mor,lre;
float e,f;
double FDA,FDP;

amp=new double[TDpoints+5];
t=new double[TDpoints+5];

infile=fopen(inPath,"r");

for(points=0;(!feof(infile))&&points<TDpoints;points++)
{ //c=e;
  fscanf(infile,"%f %f",&e,&f);
  amp[points]=f;
  t[points]=e;
//  if(points==3)
//    dt=e-c;
};
fclose(infile);
points--;

for(a=0;a<points;a++)
amp[(int)a]*=dt;

outfile=fopen(outPath,"w");
for(freq=SF;freq<=EF;freq+=df)
{ for(a=0,re=0;a<TDpoints;a++)
  { mor=complex<double>(0,-1)*freq*t[(int)a]*6.2831;
    re+=amp[(int)a]*exp(mor);
  };
  if((arg(re)-arg(lre))>4.5)
  BasePhase=BasePhase-1;
  if((arg(re)-arg(lre))<(-4.5))
  BasePhase=BasePhase+1;

  FDA=sqrt(norm(re));
  FDP=arg(re)+BasePhase*6.2831;
  fprintf(outfile,"%E %E %E\n",freq,FDA,FDP);
}

```



```

lre=re;

if(FDA>FDAMAX)
FDAMAX=FDA;
if(FDA<FDAMIN)
FDAMIN=FDA;
if(FDP>FDPMAX)
FDPMAX=FDP;
if(FDP<FDPMIN)
FDPMIN=FDP;
};
fclose(outfile);

delete amp;
delete t;
}

void __fastcall TMainForm::FOURIERTRANSFORM2Click(TObject *Sender)
//傅式轉換前置、後置作業
{ FILE *in,*FDinfo;
  int a,b;
  char infile[512],outfile[512];

  in=fopen(InfoFile.c_str(),"r");
  fscanf(in,"%d %d %d %d",&Xpoints,&Ypoints,&Xres,&Yres);
  fclose(in);
  FourierBaseDirectory=BaseDirectory+"Fourier Spectrum";
  CreateDir(FourierBaseDirectory);

  FDAMAX=0;
  FDAMIN=0;
  FDPMAX=0;
  FDPMIN=0;

  for(a=0;a<Xpoints;a++)
  for(b=0;b<Ypoints;b++)
  { sprintf(infile,"%s%d %d.txt",BaseDirectory.c_str(),a,b);
    sprintf(outfile,"%s\\%d %d.txt",FourierBaseDirectory.c_str(),a,b);

```



```

FT(infile,outfile,deltaT,FDSfreq,FDEfreq,FDdf);

};

sprintf(outfile,"%s\\Fourier.tif",FourierBaseDirectory.c_str());
FDinfo=fopen(outfile,"w");
fprintf(FDinfo,"%f %f %f\n",FDSfreq,FDEfreq,FDdf);
fprintf(FDinfo,"%d %d\n",Xpoints,Ypoints);
fprintf(FDinfo,"%f %f %f %f",FDAMAX,FDAMIN,FDPMAX,FDPMIN);
fclose(FDinfo);
}
//-----
void __fastcall TMainForm::TimeDomain1Click(TObject *Sender)//時域成像舊版，廢棄
{ int a,b,n,m;
  char intxt[128],outbmp[128];
  FILE *in;
  float t,amp;
  AnsiString temp;

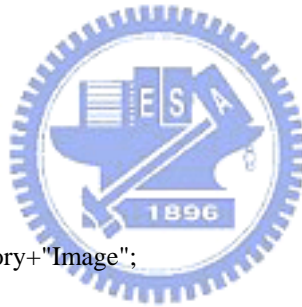
  Image1->Width=Xpoints;
  Image1->Height=Ypoints;
  ImageBaseDirectory=BaseDirectory+"Image";
  CreateDir(ImageBaseDirectory);

  IntensityPic(TDMAX,TDMIN) ;
  temp=ImageBaseDirectory+"\\\"+\"TDIntensity.bmp\"";
  Image3->Picture->SaveToFile(temp);

  for(n=0;n<TDpoints;n++)
  { for(a=0;a<Xpoints;a++)
    for(b=0;b<Ypoints;b++)
    { sprintf(intxt,"%s%d %d.txt",BaseDirectory.c_str(),a,b);
      in=fopen(intxt,"r");
      for(m=0;m<=n;m++)
      fscanf(in,"%f %f",&t,&amp);
      fclose(in);

Image1->Canvas->Pixels[a][b]=(TColor)0x010101*(int)(256*(amp-TDMIN)/(TDMAX-TDMIN));

```



```

};
temp=ImageBaseDirectory+"\\ "+n+".bmp";
Image1->Picture->SaveToFile(temp);
};
}
//-----
void __fastcall TMainForm::FrequencyDomain1Click(TObject *Sender)//頻域振幅成像
{ int a,b,n,m;
  char intxt[512],outbmp[512];
  FILE *in;
  float F,amp,phase;
  AnsiString temp;
  long ***picData;
  TImage *outputs;

  AOwner = Image1->Owner;

  picData=new long**[(int)(FDEfreq-FDSfreq)/FDdf+1];製作三維陣列
  for(n=0;n<=(FDEfreq-FDSfreq)/FDdf;n++)
  picData[n]=new long*[Xpoints];
  for(n=0;n<=(FDEfreq-FDSfreq)/FDdf;n++)
  for(a=0;a<Xpoints;a++)
  picData[n][a]=new long[Ypoints];

  FourierBaseDirectory=BaseDirectory+"Fourier Spectrum";
  FourierImageABaseDirectory=BaseDirectory+"FourierAmpImage";
  CreateDir(FourierImageABaseDirectory);

  IntensityPic(FDAMAX,FDAMIN) ;
  temp=FourierImageABaseDirectory+"\\ "+FDAMIN+".bmp";
  Image3->Picture->SaveToFile(temp);

  for(a=0;a<Xpoints;a++) //讀入各像素數據並儲存
  for(b=0;b<Ypoints;b++)
  { sprintf(intxt,"%s\\%d %d.txt",FourierBaseDirectory.c_str(),a,b);
    in=fopen(intxt,"r");
    for(n=0;n<=(FDEfreq-FDSfreq)/FDdf;n++)

```

```

    { fscanf(in,"%f %f %f",&F,&amp;,&phase);
      picData[n][a][b]=0x010101*(int)(256*(amp-FDAMIN)/(FDAMAX-FDAMIN));
    };
    fclose(in);
};

```

```

for(n=0;n<=(FDEfreq-FDSfreq)/FDdf;n++) //作圖

```

```

{ outputs = new TImage(AOwner);
  outputs->Width=Xpoints;
  outputs->Height=Ypoints;

  for(a=0;a<Xpoints;a++)
  for(b=0;b<Ypoints;b++)
  outputs->Canvas->Pixels[a][b]=(TColor)picData[n][a][b];

```

```

temp=FourierImageABaseDirectory+"\\ "+n+".bmp";

```

```

outputs->Picture->SaveToFile(temp);

```

```

delete outputs;
};

```



```

for(n=0;n<=(FDEfreq-FDSfreq)/FDdf;n++)
for(a=0;a<Xpoints;a++)
delete picData[n][a];
for(n=0;n<=(FDEfreq-FDSfreq)/FDdf;n++)
delete picData[n];
delete picData;
}

```

```

//-----

```

```

void __fastcall TMainForm::FrequencyDomainphase1Click(TObject *Sender)//頻域相位成像

```

```

{ int a,b,n,m;
  char intxt[512],outbmp[512];
  FILE *in;
  float F,amp,phase;
  AnsiString temp;
  long ***picData;

```

```

TImage *outputs;

AOwner = Image1->Owner;

picData=new long**[(int)(FDEfreq-FDSfreq)/FDdf+1];
for(n=0;n<=(FDEfreq-FDSfreq)/FDdf;n++)
picData[n]=new long*[Xpoints];
for(n=0;n<=(FDEfreq-FDSfreq)/FDdf;n++)
for(a=0;a<Xpoints;a++)
picData[n][a]=new long[Ypoints];

FourierBaseDirectory=BaseDirectory+"Fourier Spectrum";
FourierImagePBaseDirectory=BaseDirectory+"FourierPhaseImage";
CreateDir(FourierImagePBaseDirectory);

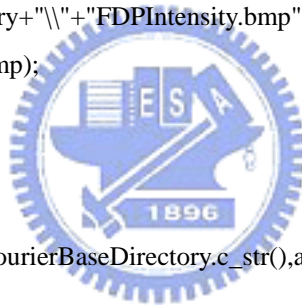
IntensityPic(FDPMAX,FDPMIN) ;
temp=FourierImagePBaseDirectory+"\\ "+FDPIntensity.bmp";
Image3->Picture->SaveToFile(temp);

for(a=0;a<Xpoints;a++)
for(b=0;b<Ypoints;b++)
{ sprintf(intxt,"%s\\%d %d.txt",FourierBaseDirectory.c_str(),a,b);
  in=fopen(intxt,"r");
  for(n=0;n<=(FDEfreq-FDSfreq)/FDdf;n++)
  { fscanff(in,"%f %f %f",&F,&A,&phase);
    picData[n][a][b]=0x010101*(int)(256*(phase-FDPMIN)/(FDPMAX-FDPMIN));
  };
  fclose(in);
};

for(n=0;n<=(FDEfreq-FDSfreq)/FDdf;n++)
{ outputs = new TImage(AOwner);
  outputs->Width=Xpoints;
  outputs->Height=Ypoints;

  for(a=0;a<Xpoints;a++)
  for(b=0;b<Ypoints;b++)
  outputs->Canvas->Pixels[a][b]=(TColor)picData[n][a][b];
};

```



```

temp=FourierImagePBaseDirectory+"\\ "+n+".bmp";

outputs->Picture->SaveToFile(temp);

delete outputs;
};
for(n=0;n<=(FDEfreq-FDSfreq)/FDdf;n++)
for(a=0;a<Xpoints;a++)
delete picData[n][a];
for(n=0;n<=(FDEfreq-FDSfreq)/FDdf;n++)
delete picData[n];
delete picData;
}
//-----
void __fastcall TMainForm::ExportIntensityPIC1Click(TObject *Sender)
{ double Max,Min;
  IntensityPic(4,-30) ;
}
//-----

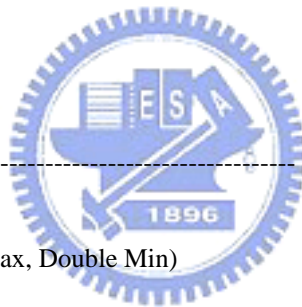
TMainForm::IntensityPic(Double Max, Double Min)
{ int a,b;
// double Max,Min;
  AnsiString out;
  char temp[20];

  for(a=8;a<32;a++)
  for(b=8;b<268;b++)
  Image3->Canvas->Pixels[a][b]=0xFF0000;

  for(a=10;a<30;a++)
  for(b=10;b<266;b++)
  Image3->Canvas->Pixels[a][b]=0x010101*(265-b);

  Image3->Canvas->Font->Size=6;
  Image3->Canvas->Font->Color=0x000080;
  for(a=0;a<=8;a++)

```



```

    { Image3->Canvas->TextOut(35,262-32*a,out.FloatToStrF(Min+(Max-Min)*a /8,0,6,3));
    };
}

void __fastcall TMainForm::Setup1Click(TObject *Sender)
{ Form1->Visible=true;
  Form1->Enabled=true;
}
//-----

void __fastcall TMainForm::LoadFrequencyDomainData1Click(TObject *Sender)
//載入頻域數據資料檔
{ FILE *inf_file;
  if(OpenDialog2->Execute())
  { InfoFile=OpenDialog2->FileName;
    BaseDirectory=InfoFile.SubString(1,InfoFile.LastDelimiter("\\")+"..\\");
    Label1->Caption=BaseDirectory;
    inf_file=fopen(InfoFile.c_str(),"r");
    fscanf(inf_file,"%f %f %f %d %d %f %f %f
%f",&FDSfreq,&FDEfreq,&FDdf,&Xpoints,&Ypoints,&FDAMAX,&FDAMIN,&FDPMAX,&FDPM
IN);
    fclose(inf_file);
    FDAMAX*=1.05;
//   FDAMIN*=1.05;
//   FDPMAX*=1.05;
//   FDPMIN*=1.05;

    FourierTransform1->Enabled=false; //關閉傅式轉換功能
    TimeDomainImages1->Enabled=false; //關閉時域相關功能
    FrequencyDomainImages1->Enabled=true; //開啟頻域相關功能

};

}
//-----

void __fastcall TMainForm::N2Click(TObject *Sender)//時域成像

```




```

{ int a,b,n,m;
  char intxt[128],outbmp[128];
  FILE *in;
  float t,amp;
  AnsiString temp;
  long ***picData;
  TImage *outputs;

  AOwner = Image1->Owner;

  picData=new long**[TDpoints];
  for(n=0;n<TDpoints;n++)
  picData[n]=new long*[Xpoints];
  for(n=0;n<TDpoints;n++)
  for(a=0;a<Xpoints;a++)
  picData[n][a]=new long[Ypoints];

  ImageBaseDirectory=BaseDirectory+"Image";
  CreateDir(ImageBaseDirectory);

  IntensityPic(TDMAX,TDMIN) ;
  temp=ImageBaseDirectory+"\\\"+\"TDIntensity.bmp\";
  Image3->Picture->SaveToFile(temp);

  for(a=0;a<Xpoints;a++)
  for(b=0;b<Ypoints;b++)
  { sprintf(intxt,"%s%d %d.txt",BaseDirectory.c_str(),a,b);
    in=fopen(intxt,"r");
    for(n=0;n<TDpoints;n++)
    { fscanf(in,"%f %f",&t,&amp);
      picData[n][a][b]=0x010101*(long)(256*(amp-TDMIN)/(TDMAX-TDMIN));
    };
    fclose(in);
  };

  for(n=0;n<TDpoints;n++)
  { outputs = new TImage(AOwner);
    outputs->Width=Xpoints;

```



```

outputs->Height=Ypoints;

for(a=0;a<Xpoints;a++)
for(b=0;b<Ypoints;b++)
outputs->Canvas->Pixels[a][b]=(TColor)picData[n][a][b];

temp=ImageBaseDirectory+"\"+n+".bmp";

outputs->Picture->SaveToFile(temp);

delete outputs;
};

for(n=0;n<TDpoints;n++)
for(a=0;a<Xpoints;a++)
delete picData[n][a];
for(n=0;n<TDpoints;n++)
delete picData[n];
delete picData;

}
//-----

```



Unit2.h

```

//-----

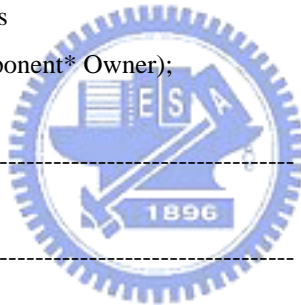
#ifndef Unit2H
#define Unit2H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>

```

```

//-----
class TForm1 : public TForm
{
__published:    // IDE-managed Components
    TLabel *Label1;
    TEdit *Edit1;
    TLabel *Label2;
    TEdit *Edit2;
    TLabel *Label3;
    TEdit *Edit3;
    TButton *Button1;
    TButton *Button2;
    void __fastcall Button1Click(TObject *Sender);
    void __fastcall Button2Click(TObject *Sender);
    void __fastcall FormShow(TObject *Sender);
private:       // User declarations
public:        // User declarations
    __fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif

```



Unit2.cpp

```

//-----

#include <vcl.h>
#pragma hdrstop

#include "Unit2.h"
#include "Unit1.h"
//-----

#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;

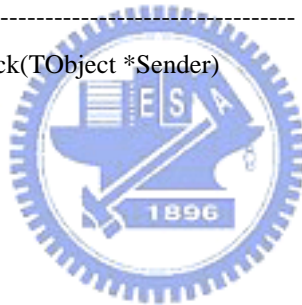
```

```

//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
    Edit1->Text=MainForm->FDSfreq;
    Edit2->Text=MainForm->FDEFreq;
    Edit3->Text=MainForm->FDdf;
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender) //設定傳式轉換參數
{
    MainForm->FDSfreq=Edit1->Text.ToDouble();
    MainForm->FDEFreq=Edit2->Text.ToDouble();
    MainForm->FDdf=Edit3->Text.ToDouble();

    Form1->Enabled=false;
    Form1->Visible=false;
}
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    Form1->Enabled=false;
    Form1->Visible=false;
}
//-----
void __fastcall TForm1::FormShow(TObject *Sender)
{
    Edit1->Text=MainForm->FDSfreq;
    Edit2->Text=MainForm->FDEFreq;
    Edit3->Text=MainForm->FDdf;
}
//-----

```



參考文獻：

- [1] 王炯翰 “液晶次毫米波段光學特性之研究(I)”,交通大學光電所碩士論文,2002
- [2] Daniel M. Mittleman, Rune H. Jacobsen, and Martin C. Nuss, “T-Ray Imaging”, JOURNAL OF SELECTED TOPICS IN QUANTUM ELECTRONICS, VOL. 2, NO. 3, SEPTEMBER 1996
- [3] 游博文 “THz 輻射偵測用砷離子佈植砷化鎵偶極天線之研製”,交通大學光電所碩士論文,2002
- [4] 洪勝富, 齊正中, ”短脈衝雷射激發兆赫輻射技術及其研究”, 物理雙月刊二三卷, pp.322-328,2001
- [5] H.BB,NUSS MC “Imaging with terahertz waves”,Opt. Letter.,vol.20, No. 16, pp. 1716,1995
- [6] P.H. Bolivar, M Brucherseifer, M, Nagel,H Kurz, A Bosserhoff, R Buttner, “Label-free probing of the binding state of DNA by time-domain terahertz sensing” Ultra. Pheno. Semi 2001,vol.384, pp.253-258,2002
- [7] X.C.Zhang, B.B. Hu, and D.H. Auston, “Generation of femtosecond electromagnetic pulses from semiconductor surface”, Appl. Phys. Lett. Vol.56 , No. 11,pp.1011-1013,1990
- [8] S.L. Chuang, S. Schmitt, “Optical rectification at semiconductor surfaces”, Phys. Rev. Lett. ,vol.68, pp.102-105 , 1992
- [9] Q. Chen, Zhiping Jiang, G. X. Xu, X.-C. Zhang,” Near-field terahertz imaging with a dynamic aperture”, OPTICS LETTERS , Vol. 25, No. 15 ,pp. 1122-1124, 2000

- [10] D.C. Edelstein, R.B. Romney, and M. Scheuermann, "Rapid programmable 300 ps optical delay scanner and signal-averaging system for ultrafast measurements", *Rev. Sci. Instrum.* 62(3),1991
- [11] M.J. Feldstein, P. Vöhringer, N. F.Scherer , "Rapid-scan pump-probe spectroscopy with high time and wave-number resolution: optical-Kerr-effect measurements of neat liquids ", *J. Opt. Soc. Am. B* , vol. 12,No. 8,1995
- [12] John M. Khosrofian and Bruce A. Garetz , "Measurement of a Gaussian laser beam diameter through the direct inversion of knife-edge data", *applied optics*, vol.22,No.21,pp.3406,1983
- [13] Lionel Duvillaret, Stéphane Riolland, and Jean-Louis Coutaz, "Electro-optic sensors for electric field measurement. II. Choice of the crystals and complete optimization of their orientation ", *J. Opt. Soc. Am. B*,vol. 19,No 11,2002
- [14] N.C.J. van der Valk, and P.C.M. Planken, "Electro-Optic detection of subwavelength terahertz spot sizes in the near field of a metal tip", *Appl. Phys. Lett.*, Vol. 81, No. 9, pp. 1558, 2002
- [15] Karsten J. Siebert, Holger Quast, Rainer Leonhardt, Torsten Löffler, Mark Thomson, Tobias Bauer, and Hartmut G. Roskos, "Continuous-wave all-optoelectronic terahertz imaging", *Appl. Phys. Lett.*, Vol.80, No. 16, pp.3003,2002
- [16] John O'Hara, and D. Grischkowsky,"Synthetic phased-array terahertz imaging",*opt. Lett.*, Vol. 27, No. 12, pp. 1070, 2002
- [17] Ajay Nahata, James T. Yardley, Tony F. Heinz, "Two-dimensional imaging of continuous-wave terahertz radiation using electro-optic detection", *Appl. Phys. Lett.*, Vol. 81, No. 6, pp. 963,2002
- [18] K. McClatchey, M.T. Reiten, and R.A. Cheville,"Time resolved synthetic aperture terahertz impulse imaging",*Appl. Phys. Lett.*, Vol. 79, No. 27, pp. 4485, 2001