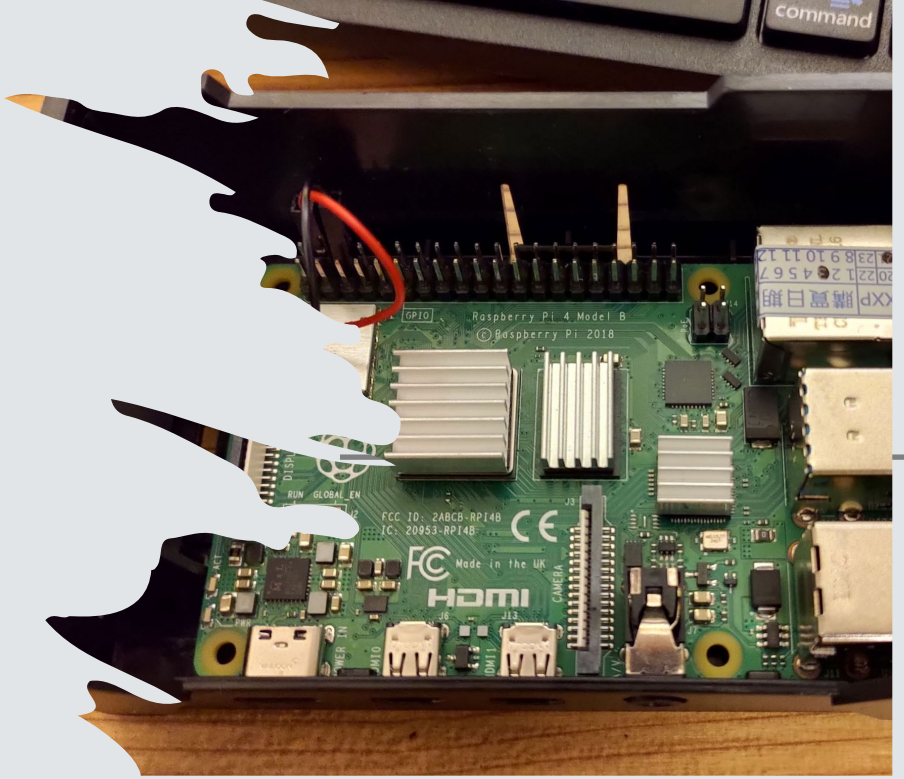
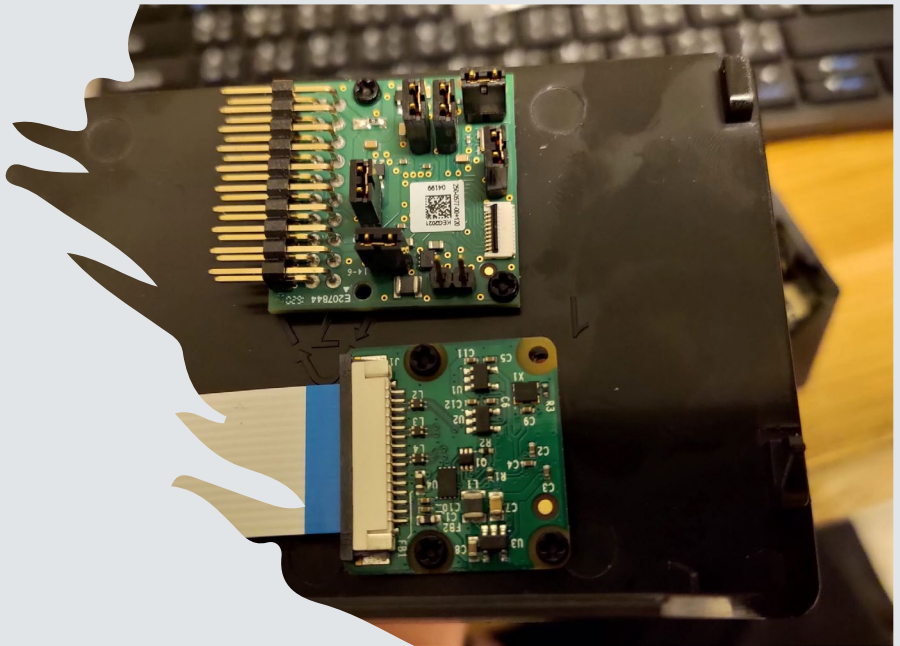
# 熱像儀

# Outline

1. 硬體
2. 安裝
3. 測試
4. 應用



# CC (Creative Commons)

## 姓名標示 — 非商業性 — 相同方式分享

姓名標示 — 你必須給予 適當表彰、提供指向本授權條款的連結，以及 指出（本作品的原始版本）是否已被變更。你可以任何合理方式為前述表彰，但不得以任何方式暗示授權人為你或你的使用方式背書。

非商業性 — 你不得將本素材進行商業目的之使用。

相同方式分享 — 若你重混、轉換本素材，或依本素材建立新素材，你必須依本素材的授權條款來散布你的貢獻物。


本教材改編自台灣樹莓派–用樹莓派自製防疫熱像

# Outline

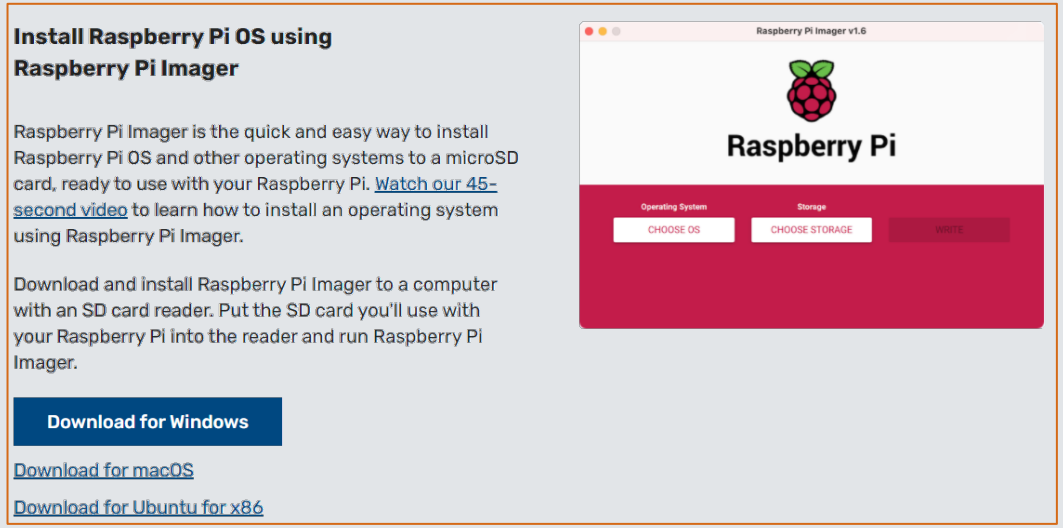
A. 課前，必要軟硬體安裝
    1. 下載映像檔 以及 燒錄到SD卡
    2. 樹莓派的基礎知識 與 硬體安裝
        ① 安裝樹莓派
        ② 相機安裝
        ③ FLIR安裝
    3. 開機後設定SSH連線以及VNC連線
    4. 使用MobaXterm來進行SSH與VNC連線
    5. 安裝QT與部分軟體
B. 原理講解
C. 實作測試
D. OpenCV 安裝

# 下載映像檔

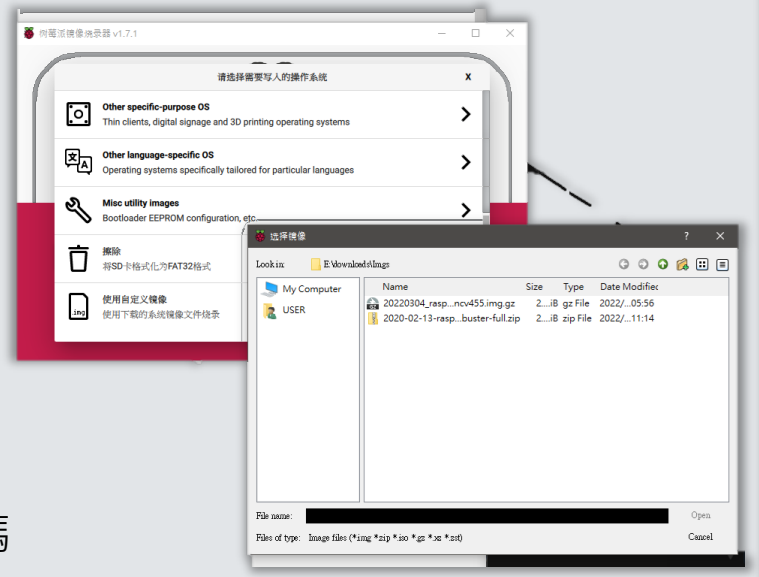
- 映像檔網址：連結

https://downloads.raspberrypi.org/raspbian_full/images/raspbian_full-2020-02-14/2020-02-13-raspbian-buster-full.zip

- 下載映像檔的同時，下載 樹莓派官方的燒錄程式 ： Raspberry Pi Imager
- 請根據自己的操作系統下載對應的程式進行安裝



# 燒錄映像檔到SD卡




- 插入SD卡到電腦讀卡機
- 開啟 **Raspberry Pi Imager**
- **[Ctrl+Shift+X]** 開啟進階設定
- 設置自己的 **hostname (主機名)**
- 開啟 **ssh**
  - 設置 **ssh** 連線時使用的名稱及密碼
- 設置樹莓派要連線的 **Wi-Fi ap** 帳號密碼及國家
- 儲存設定
- **CHOOSE OS > Use custom >** 選擇剛剛下載的**img**檔
- **CHOOSE CARD >** 選擇SD卡
- **WRITE**

**Wi-Fi** 的頻段必須是 **2.4GHz** 才可以直接連上

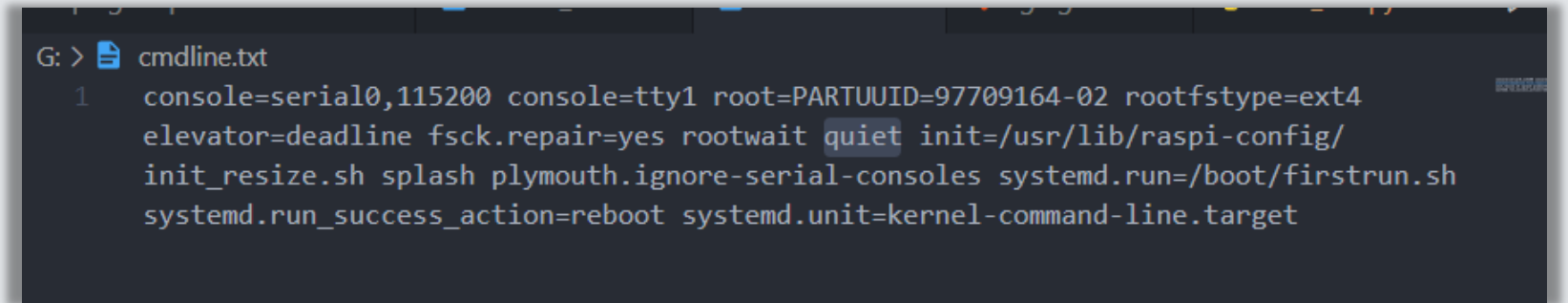
# 燒錄後設定

完成燒錄後會看到右邊的圖


接下來請在 SD卡中 /boot/config.txt 檔案的最

```
[thermal]
dtoverlay    = pi3-miniuart-bt
core_freq    = 250
enable_uart = 1
```

接下來請修改SD卡中的 /boot/cmdline.txt 檔案，將 quiet 移除



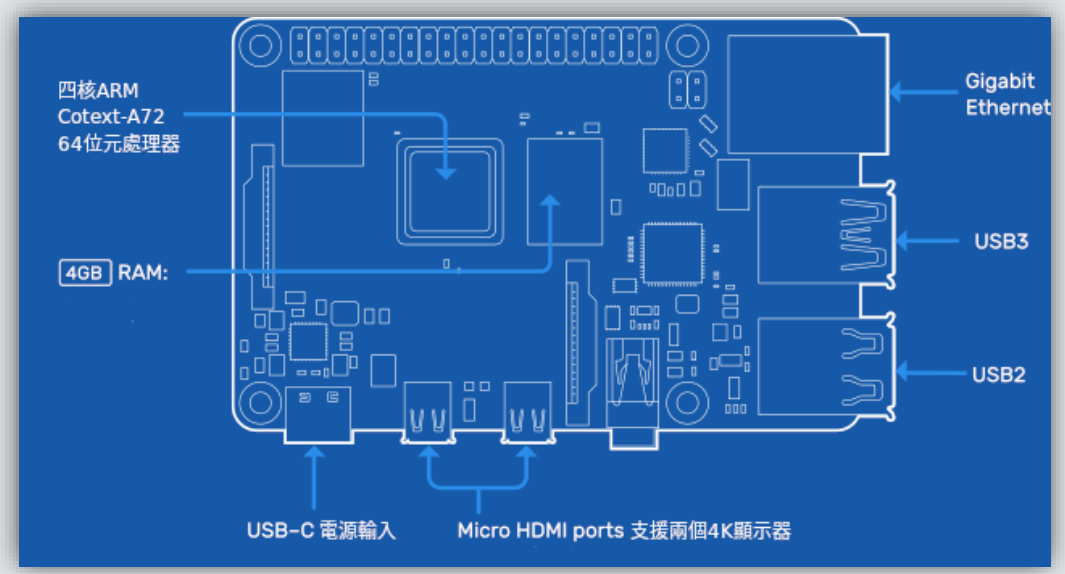
# 樹莓派的基礎知識

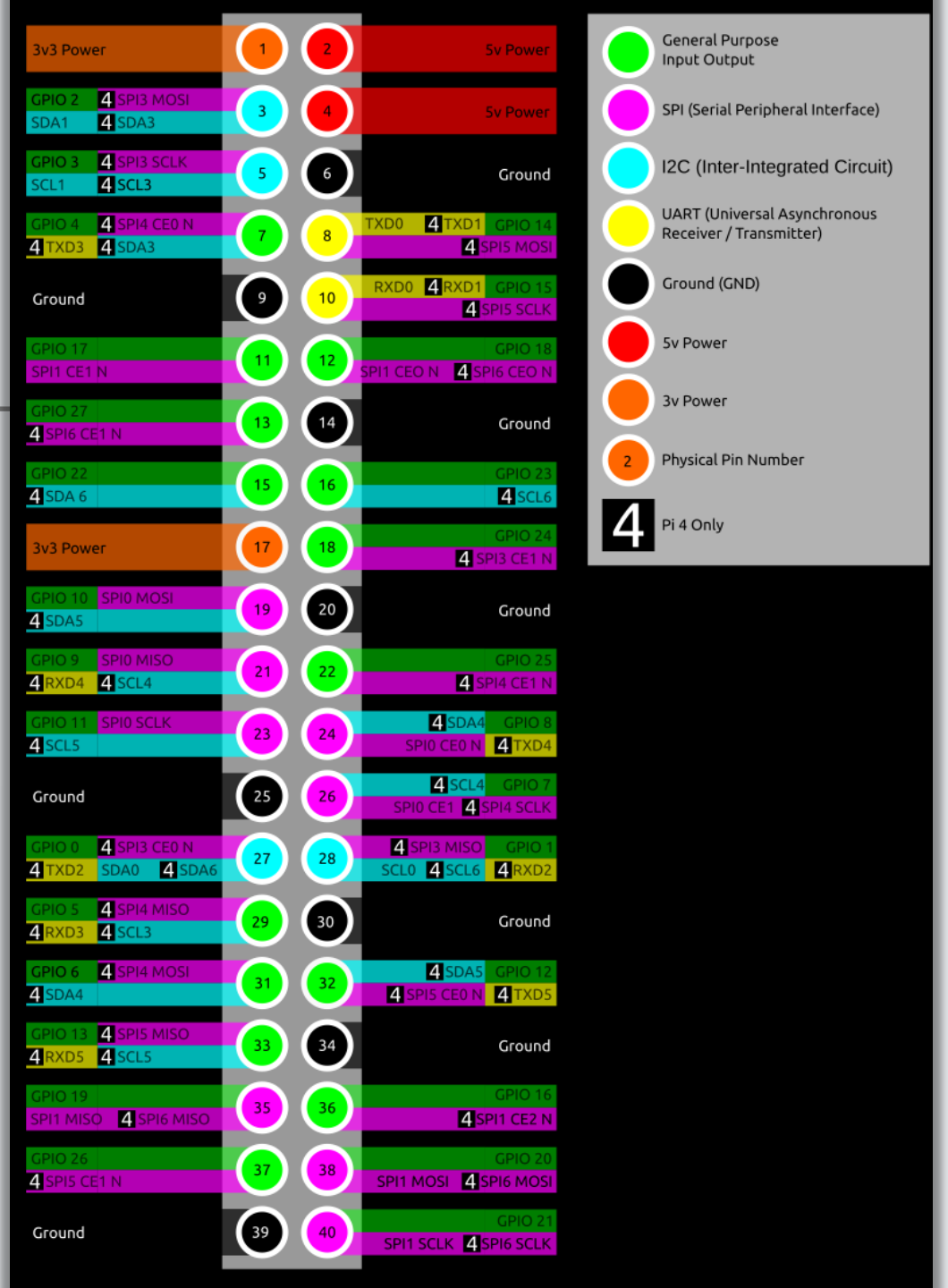
一台小型的電腦

使用記憶卡作為儲存空間

通常使用 `linux` 系統

本課程使用 [raspberry pi 4 B]



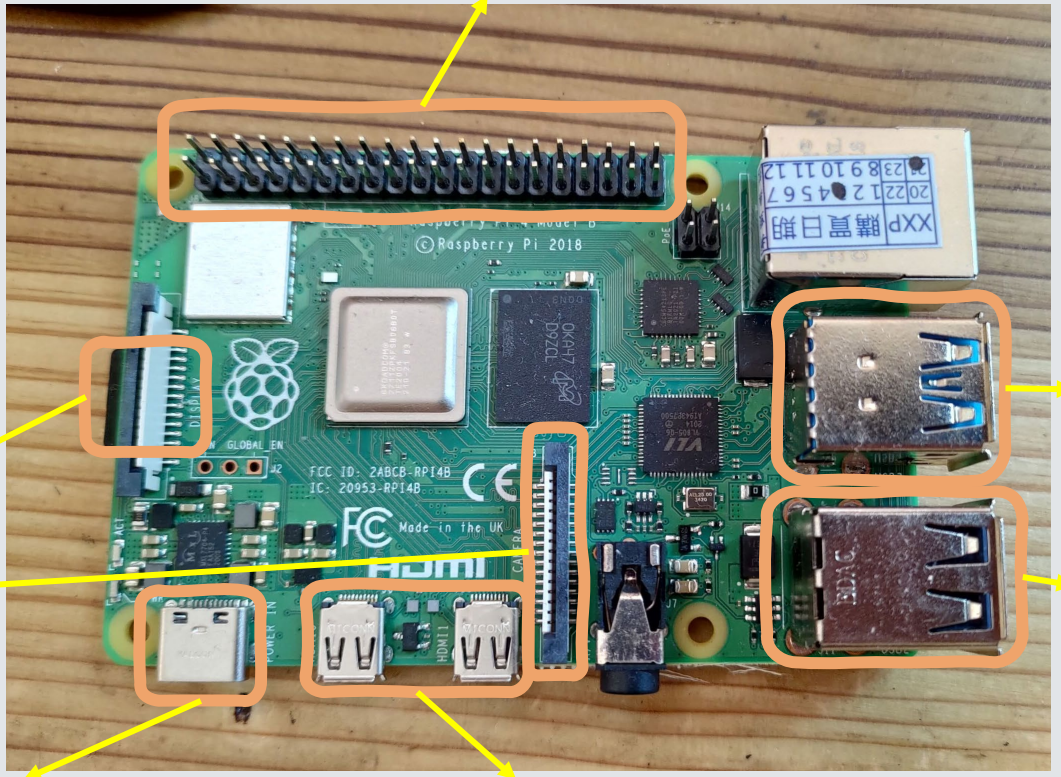
# 硬體安裝 – 樹莓派相機 與 SD 卡

切記，所有硬體操作都需要在斷電下進行

將剛剛燒錄好的SD卡插入樹梅派
螢幕的 **micro-HDMI** 線優先使用左側的埠
相機的排線插入圖中相機安裝位置
(排線上藍色貼紙向右)

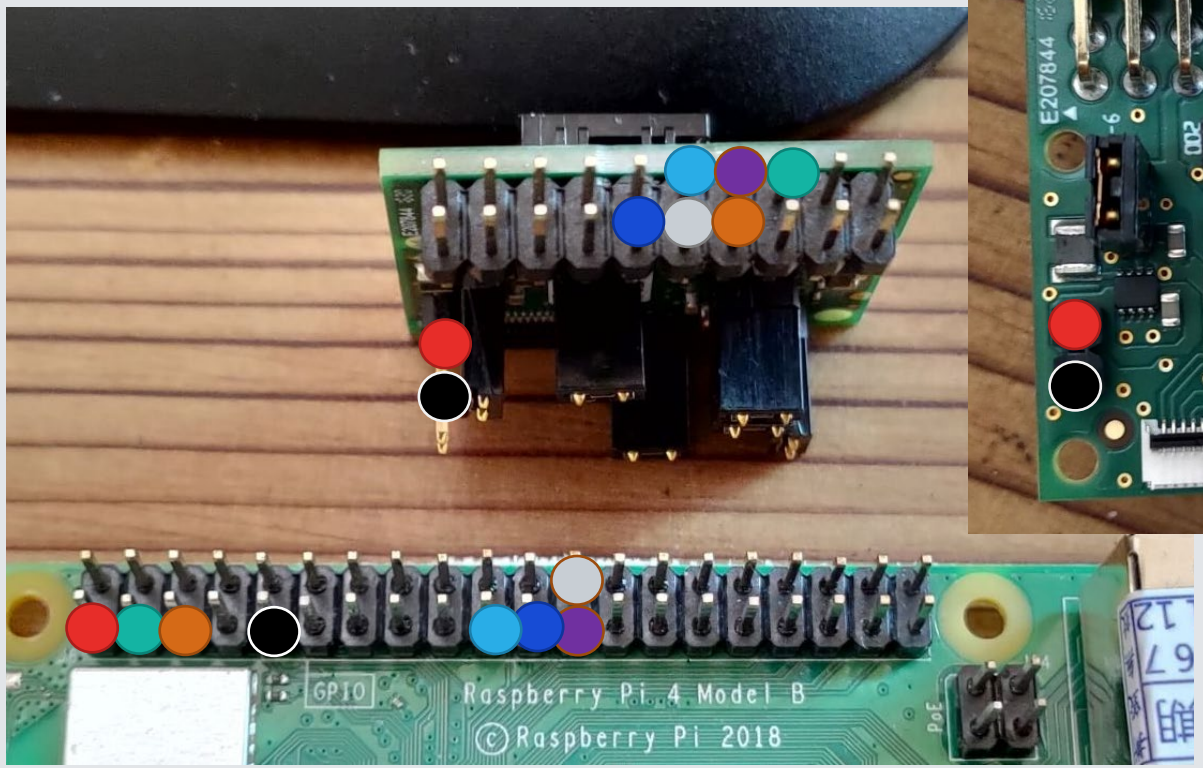
等所有東西都裝好才插上電源

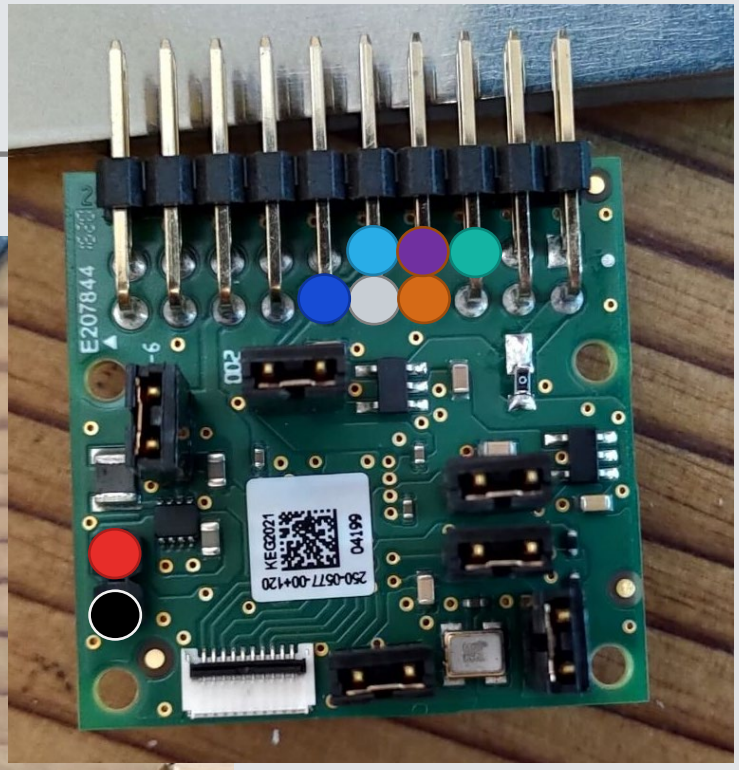
**GPIO** 也是等一下要連接 **FLIR** 的位置



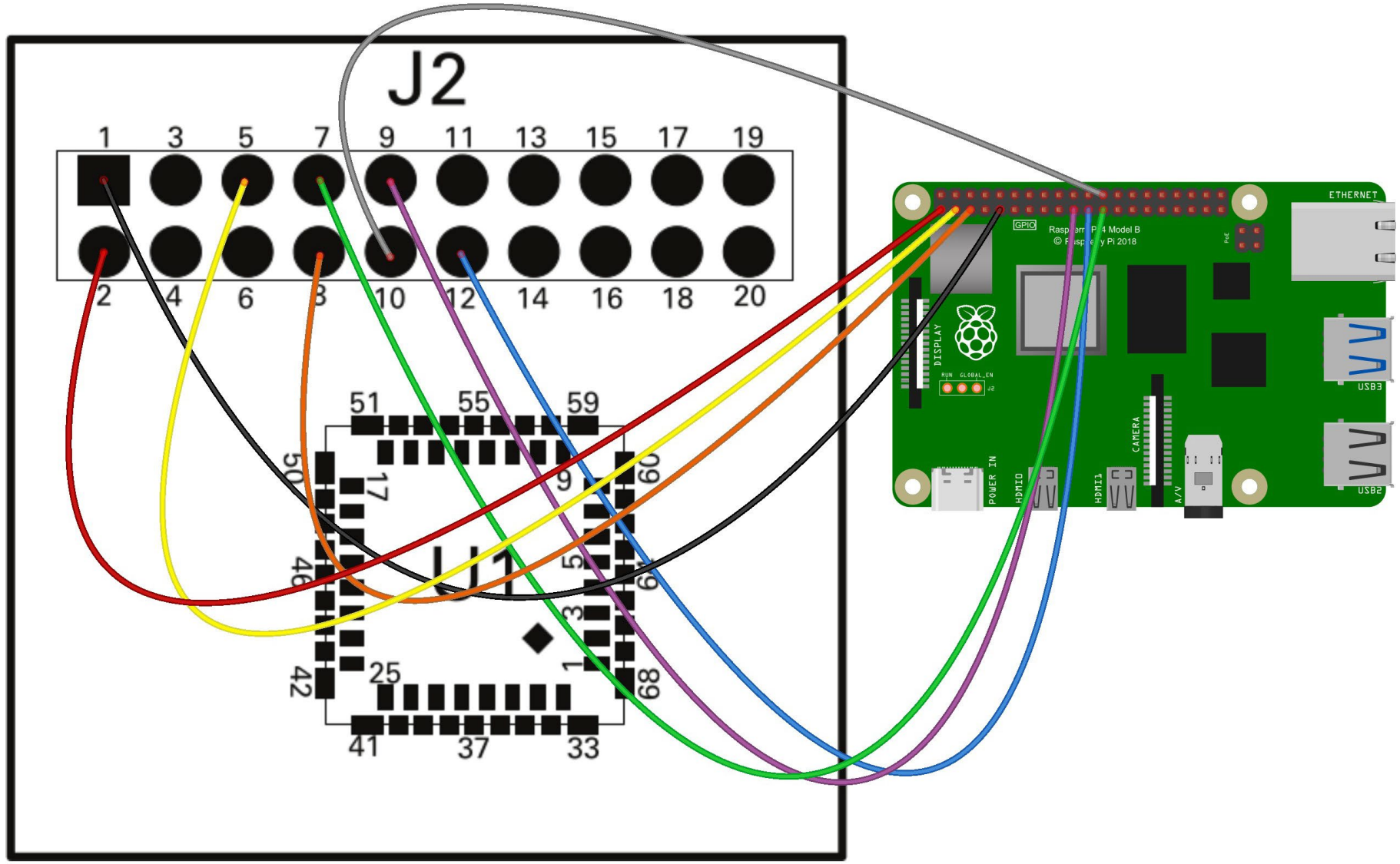
# 硬體安裝 – FLIR Lepton 3.5

- 5v 電源
- GND
- GPIO 2
- GPIO 3
- SPI0 MOSI
- SPI0 MISO
- SPI0 SCLK
- SPI0 CEO N



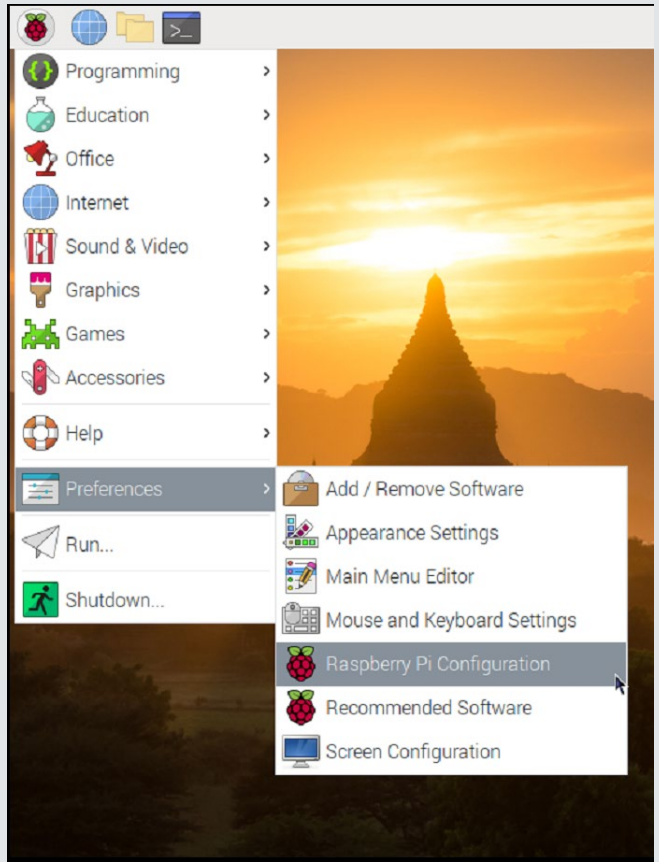



舊版如圖所示，與新版幾乎一樣除了電源接地，
新版的電源與接地在開發版中間偏左

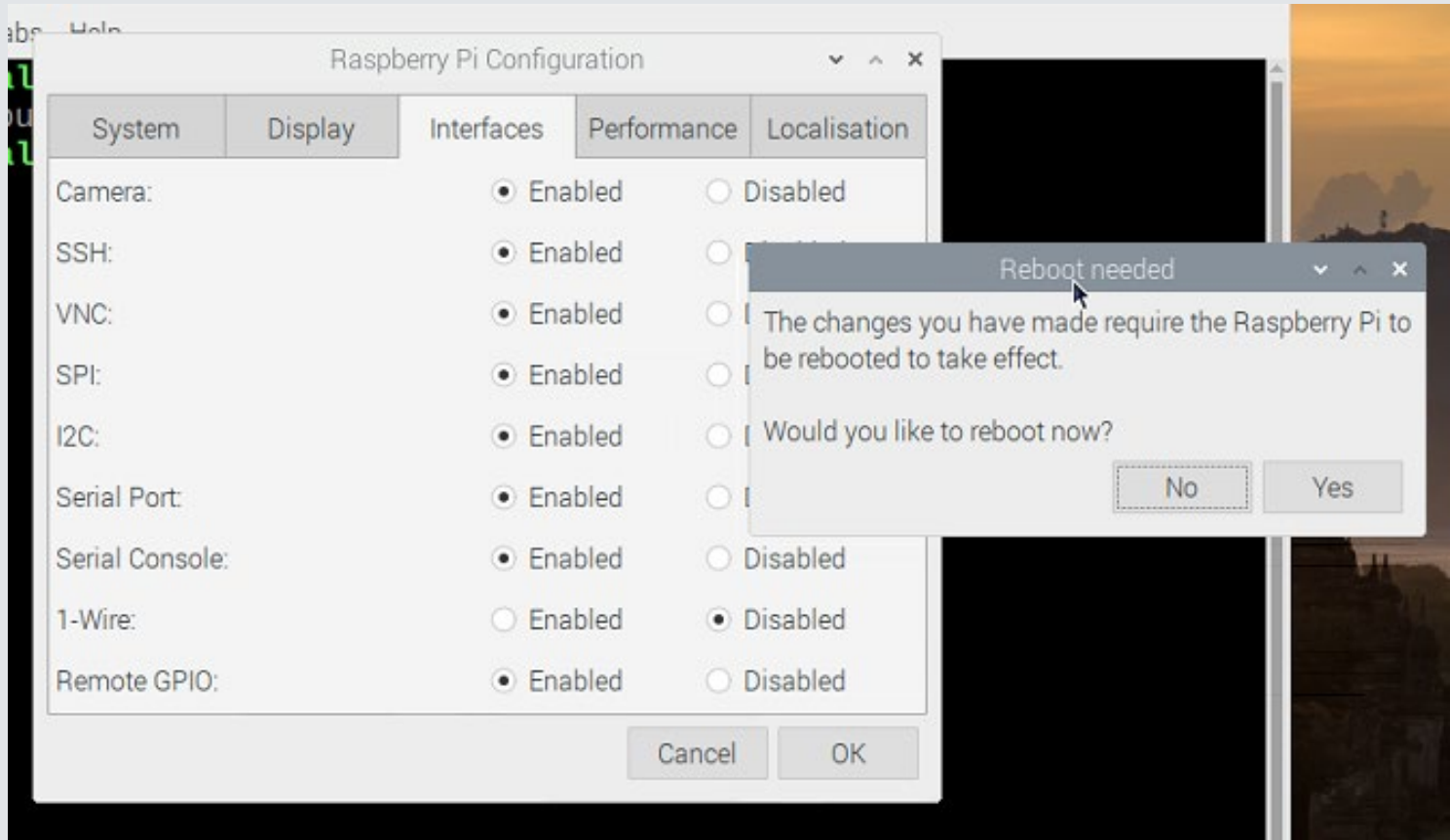


# 開機後設定-啟用 interfaces

打開選單裡的 樹莓派設定



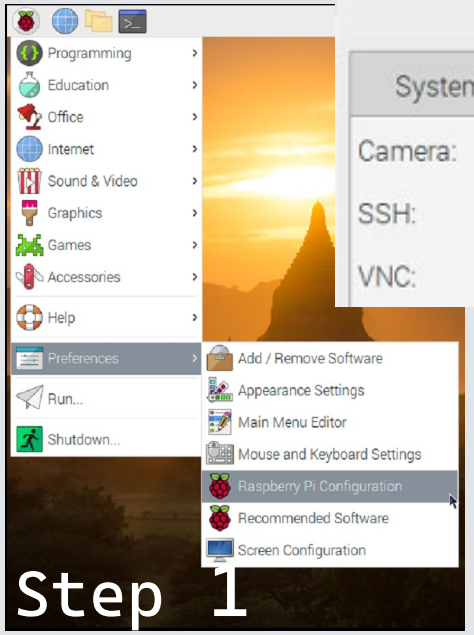

將 **interfaces** 分頁依照以下項目打勾

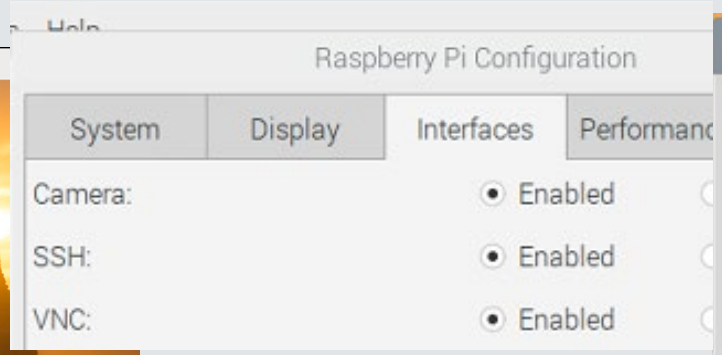


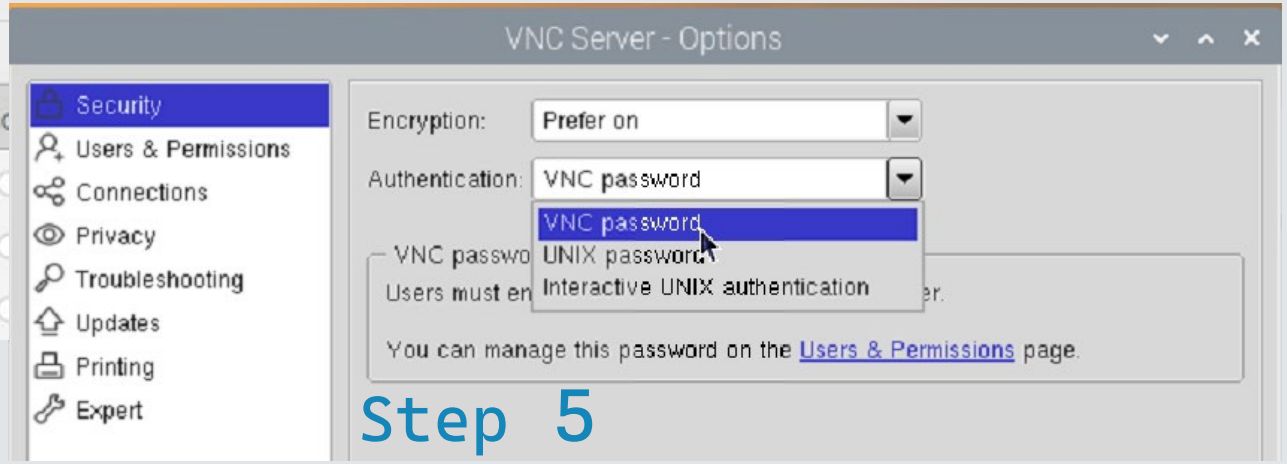
# 開機後設定-啟用 傳統 VNC

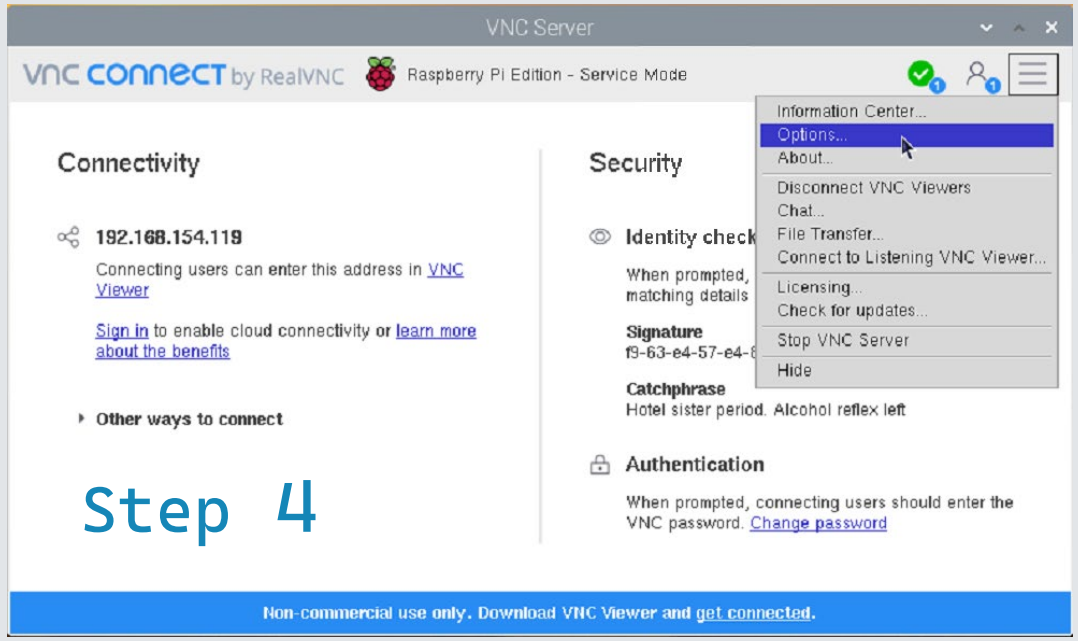
1. 打開選單裡的 樹莓派設定
2. 打開 interfaces 中的 VNC，後按 OK
3. 按開右上角的 VS ( VNC Server )
4. 打開選單中的選項
5. 選擇 安全 分頁，在裡面驗證方法選 VNC Passwd
6. 選擇 使用者及權限 分頁，選擇 標準使用者 後 按右邊的 password ，在彈出的對話框中輸入你的 VNC 密碼，然後一路按 OK
7. 最後用你的 VNC 軟體就可以連接上主頁顯示的 IP 了

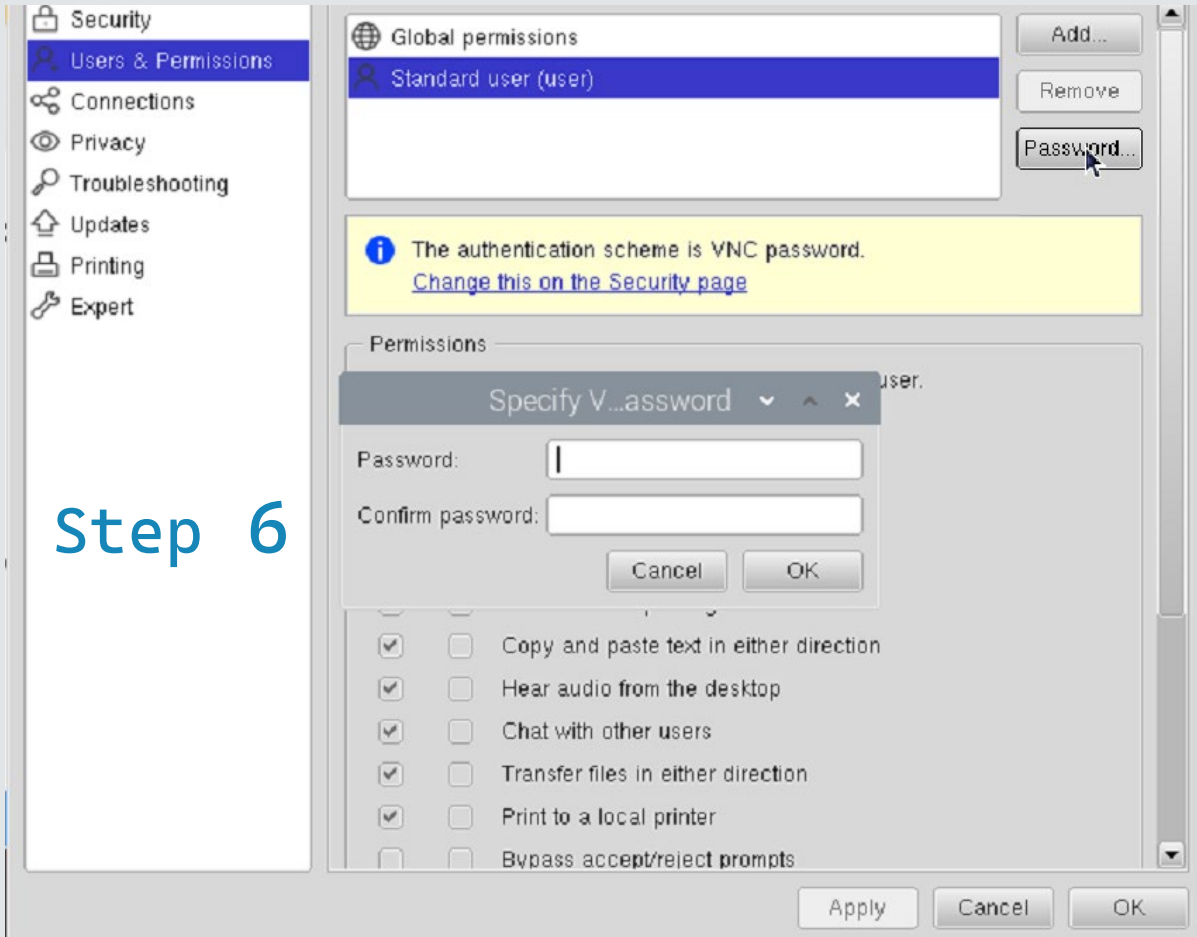
# 開機後設定-啟用傳統 VNC



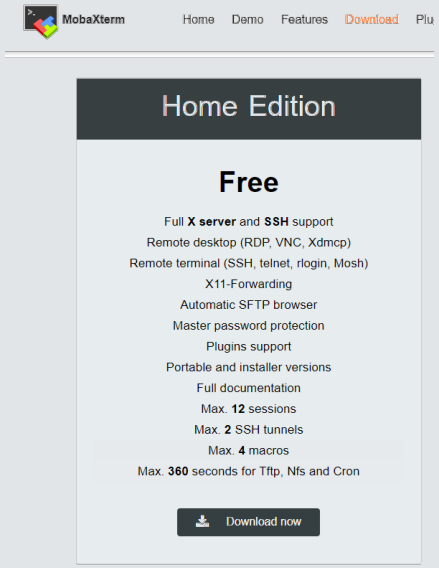



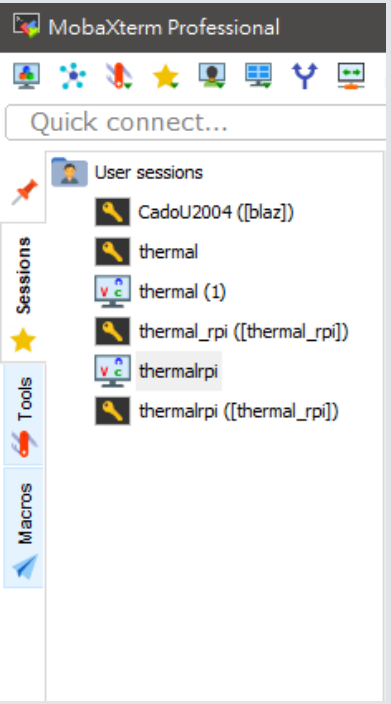
# 使用MobaXterm來進行SSH與VNC連線

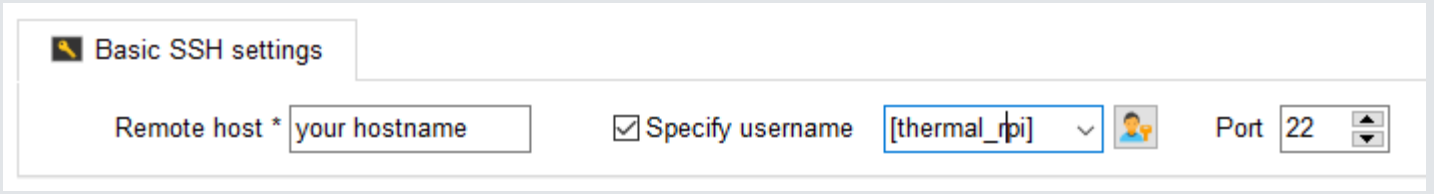
1. 先確定你的電腦與樹莓派在同一網路下，
   1. 例如以 **wifi** 或有線插入到同一台分享器/數據機/AP
   2. 打開網路供應設備查看是否可以看到你設定的主機名
2. 筆電下載並安裝 MobaXterm 網址(https://mobaxterm.mobatek.net/download.html)

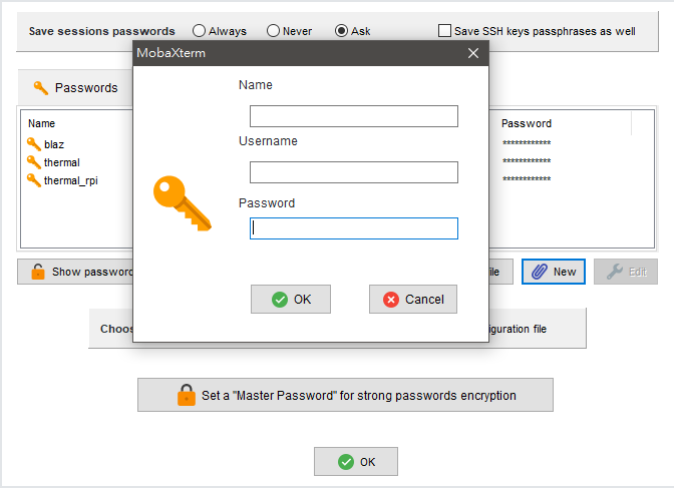



# 使用MobaXterm來進行SSH與VNC連線

1. 打開 MobaXterm 開啟新 session (ctrl + shift + N)
2. 選擇 SSH 分頁



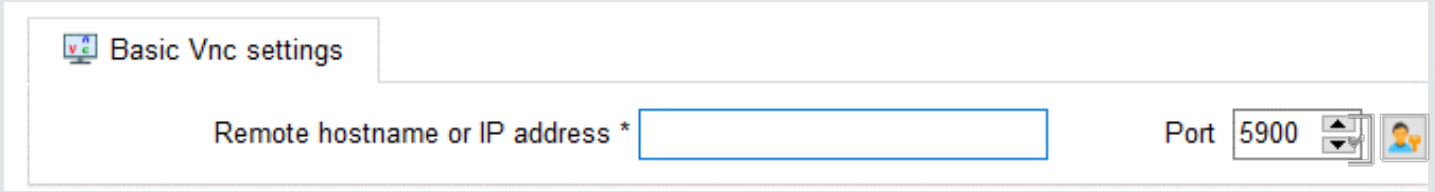



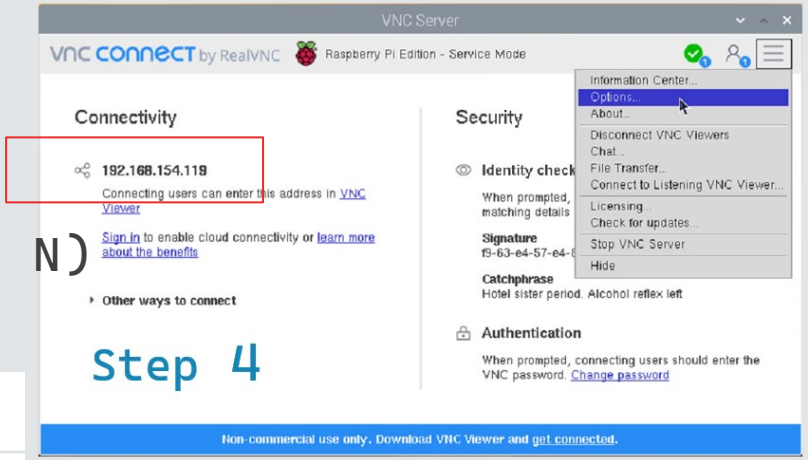




1. **remote host** 填入你設定的主機名或**RPI's IP**
2. 想要儲存登入資訊，可以勾選 **specify username** 在右邊的  按下去
    1. 按下 **new** 之後填入識別名稱、**rpi** 的使用者名稱、**rpi** 的使用者密碼，之後按下兩次 **OK**
    2. 在 **specify username** 選擇框裡選擇剛剛創建的帳號密碼
3. 最後按下 **OK** ，此時在側欄 **Sessions** 處應該就會出現剛剛創建的 **session** 點他就可以連線了

# 使用MobaXterm來進行VNC連線

1. 先確定樹莓派已經開啟 VNC (會在狀態欄顯示　　圖示)
2. 確保已經照前面投影片開啟傳統 VNC
3. 打開 MobaXterm 開啟新 session (ctrl + shift + N)
4. 選擇 VNC 分頁



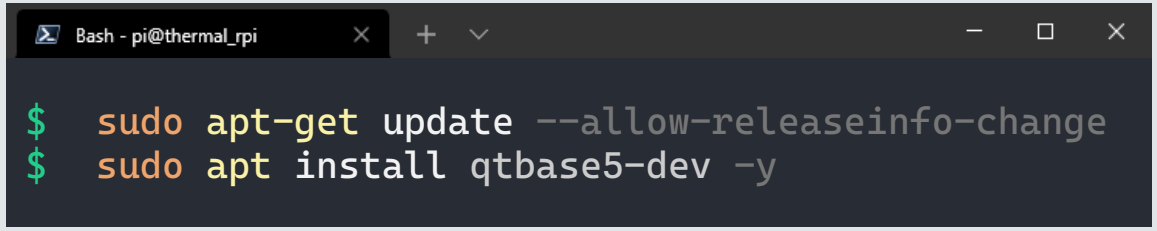




1. 在 Remote hostname or IP address 填入 IP ( 按樹莓派狀態列的VNC圖示會看到紅框處 IP )
2. 最後按下 OK ，此時在側欄 Sessions 處應該就會出現剛剛創建的 session 點他就可以連線了

# 安裝 QT 與 部分軟體

由於 QT4 已經被 QT5 替換不再提供下載，所以我們改裝 QT5

由於我們使用的系統已經過時，因此更新需附加上參數 `--allow-releaseinfo-change`



# Outline

A. 課前，必要軟硬體安裝
B. 原理講解
    1. 前言
    2. 紅外線測溫原理
    3. 紅外線相機介紹
C. 實作測試
D. OpenCV 安裝

# 2020 生活日常


https://news.ltn.com.tw/news/life/paper/1364790

# 2020 生活日常


https://www.ftvnews.com.tw/news/detail/2020327L05M1

# 使用紅外線量測溫度


https://lepton.flir.com/wp-content/uploads/2015/06/Introduction-to-Infrared-Thermography-Basics.pdf

# 為什麼需要紅外線影像？

- 我們需要外部的光源來讓我們”看到”
- 紅外線影像讓我們對溫度有”第六感”

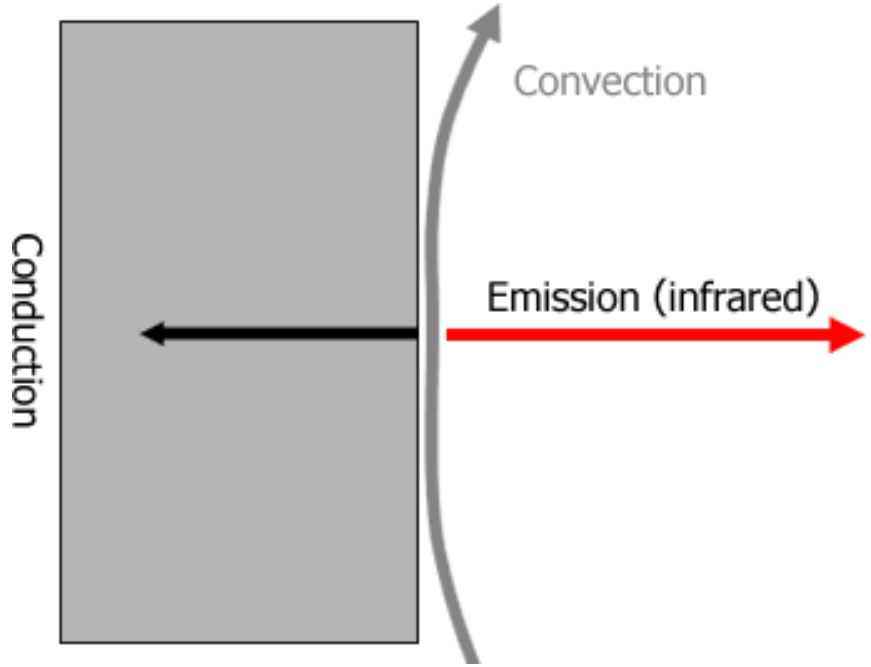

可見光影像受霧霾影響

紅外線影像不受天候或光線影響

https://www.axis.com/en-us/products/thermal-cameras

# 熱傳遞方法

- 對流 (Convection)：經由流體傳遞
- 傳導 (Conduction)：經由固體傳遞
- 輻射 (Emission)：不須經過介質傳遞 ( 光波形式 )



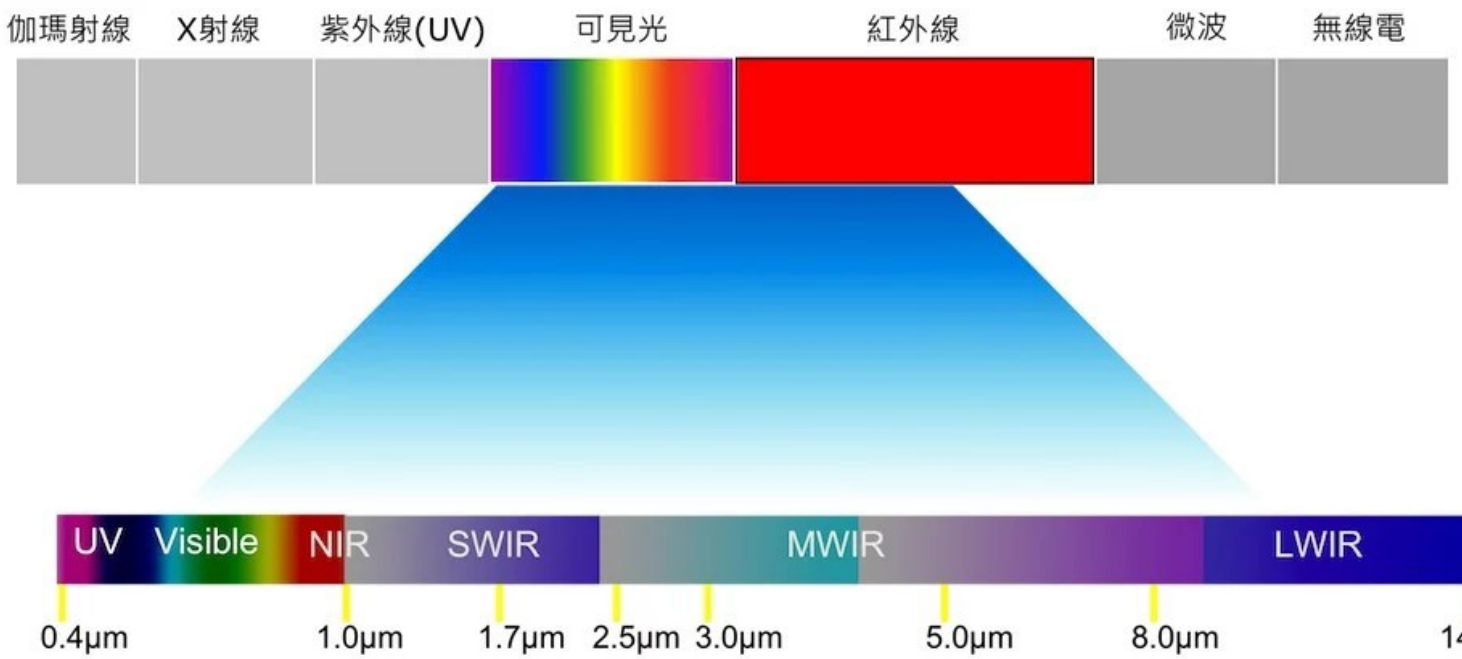

http://www.infrared.avio.co.jp/en/products/ir-thermo/what-thermo.html

# 紅外線輻射 (Radiometric Infrared)

- 絕對溫度超過 (-273℃) 的所有物體都會發出紅外線輻射
- 輻射傳播不需介質，不受白天晚上、煙霧、下雨等影響
- 紅外線輻射能量由物體的溫度和發射決定
  - 從 0.76μm (紅光邊緣) 到 1000μm (微波範圍起點)
- SWIR:0.9-1.7μm
  - 短波紅外線，適用於戶外夜間拍攝
- MWIR:3-5μm
  - 中長波紅外線，適用遠端監測，常用在低溫熱像檢測
- LWIR:8-14μm
  - 長波紅外線，適用多塵有霧環境，常用在室溫熱像檢測

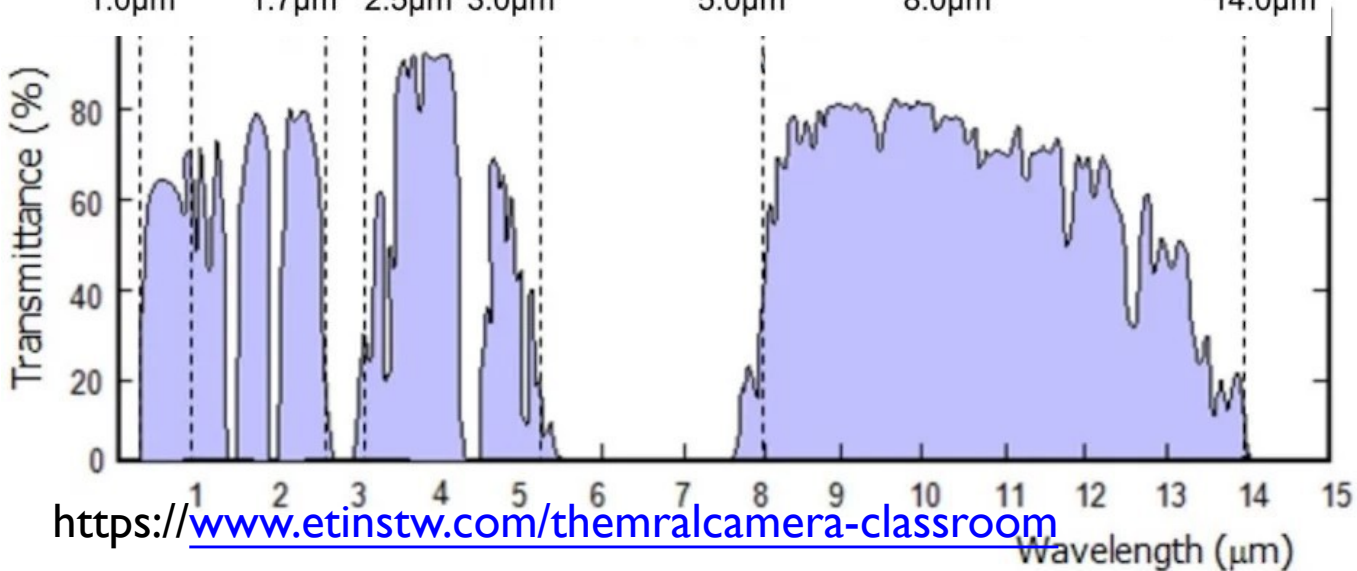
https://www.opgal.com/blog/thermal-cameras/the-physics-behind-thermal-imaging

# 紅外線波長










https://www.etinstw.com/themralcamera-classroom

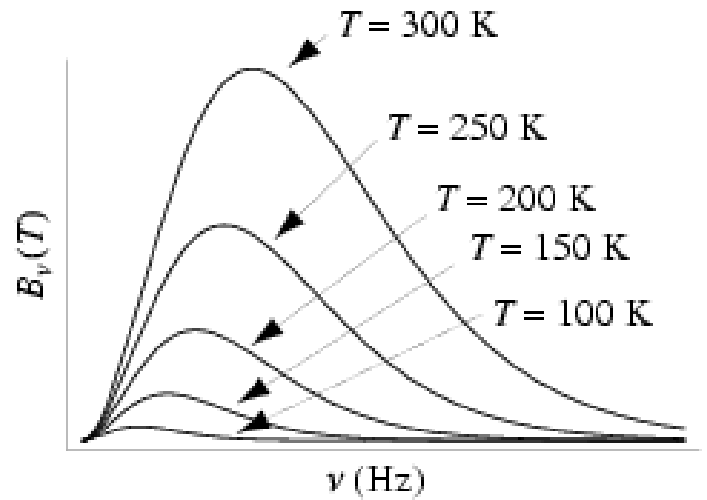
# 四大定律

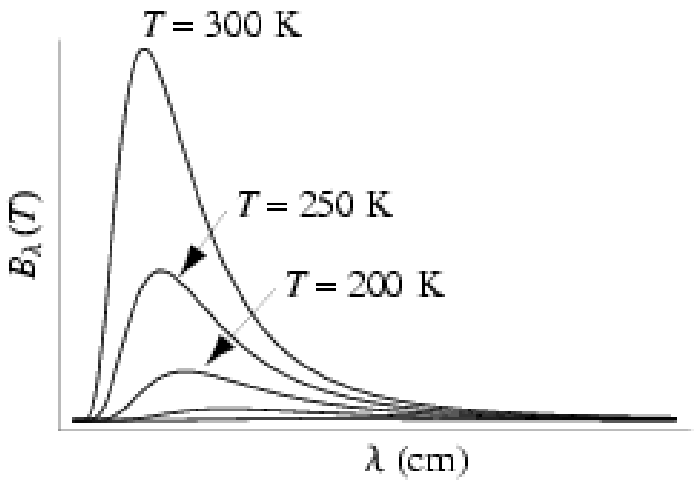
- Planck's law
  - 一個黑體發射出的電磁輻射的輻射率與頻率關係
- Wien's displacement law
  - 黑體輻射光譜輻射度的峰值波長與自身溫度成反比
- Stefan-Boltzmann law
  - 黑體輻射出的總能量與溫度T的四次方成正比
- Kirchhoff's law of thermal radiation
  - 描述真實物體的發射率與吸收率之間的關係

http://www.infrared.avio.co.jp/en/products/ir-thermo/what-thermo.html

# Planck's Law

- 一個黑體發射出的電磁輻射的輻射率與頻率關係



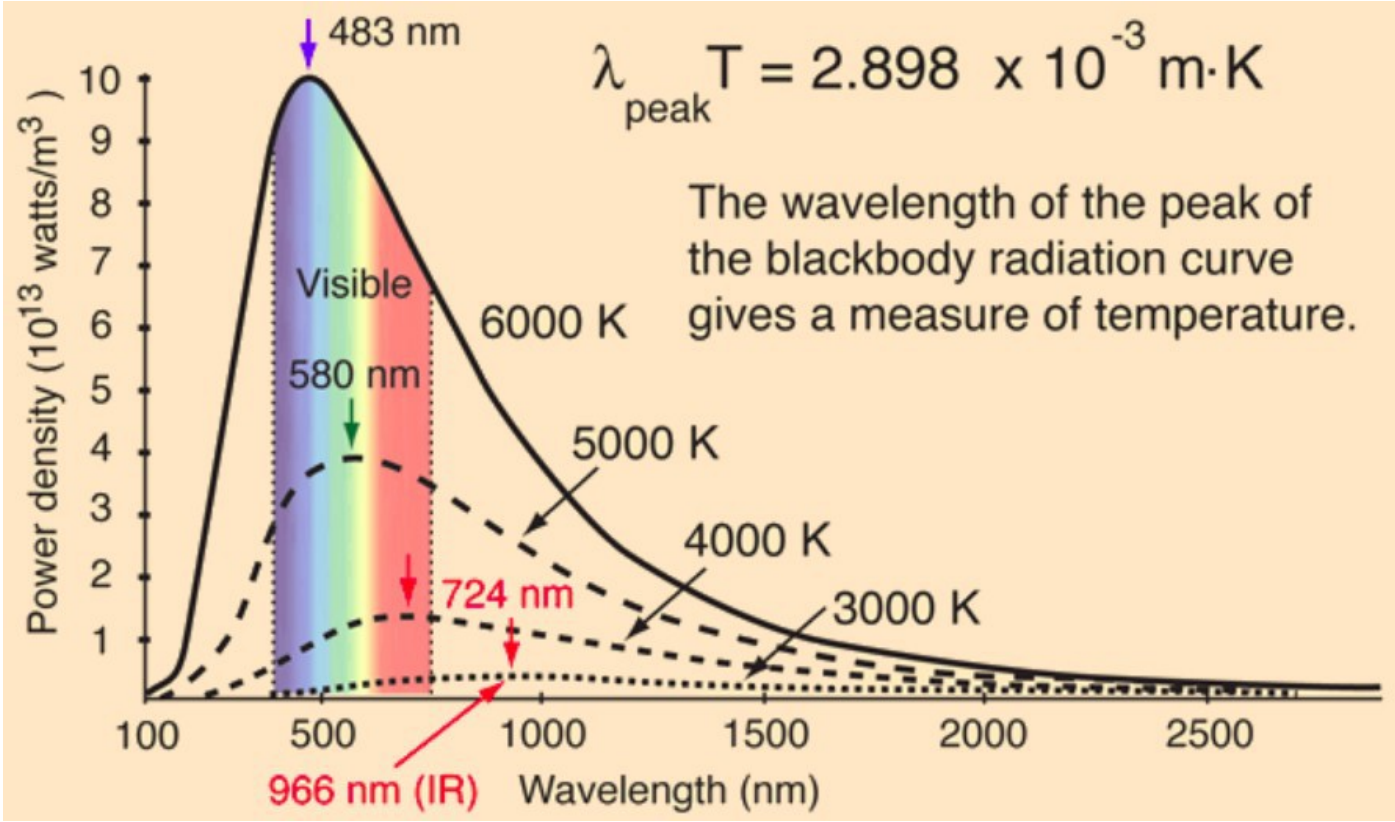

$$I_\nu(\nu,T)=\frac{2h\nu^3}{c^2}\frac{1}{e^{\frac{h\nu}{kT}}-1}$$

$$I_\lambda(\lambda,T)=\frac{2hc^2}{\lambda^5}\frac{1}{e^{\frac{hc}{\lambda kT}}-1}$$

https://scienceworld.wolfram.com/physics/PlanckLaw.html

# Wien's Displacement Law

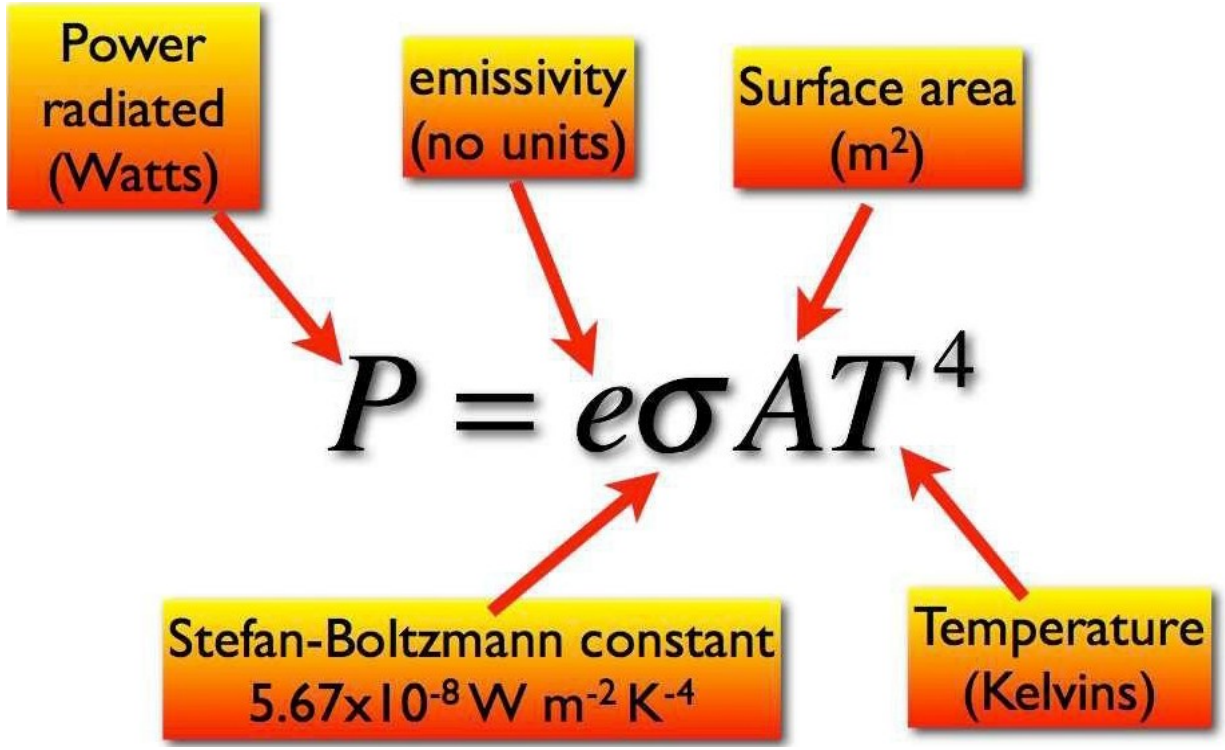
- 黑體輻射光譜輻射度的峰值波長與自身溫度成反比




http://hyperphysics.phy-astr.gsu.edu/hbase/wien.html

# Stefan-Boltzmann Law

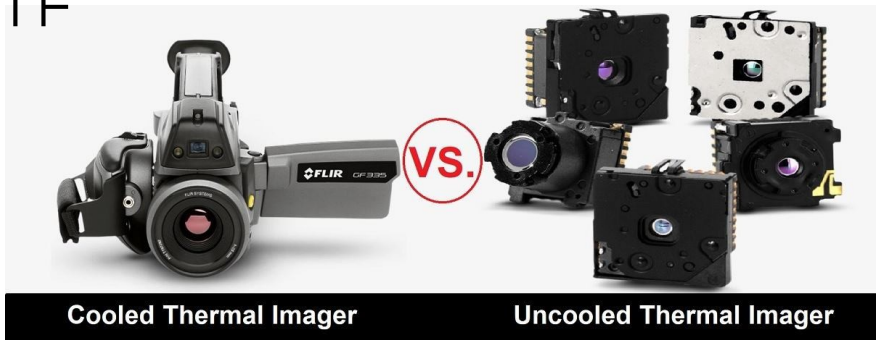
- 黑體每單位時間在所有波長上每單位表面積輻射的總能量與黑體熱力學溫度 T 的四次方成正比




https://www.deviantart.com/zackdowney/art/Stefan-Boltzmann-Law-407403124

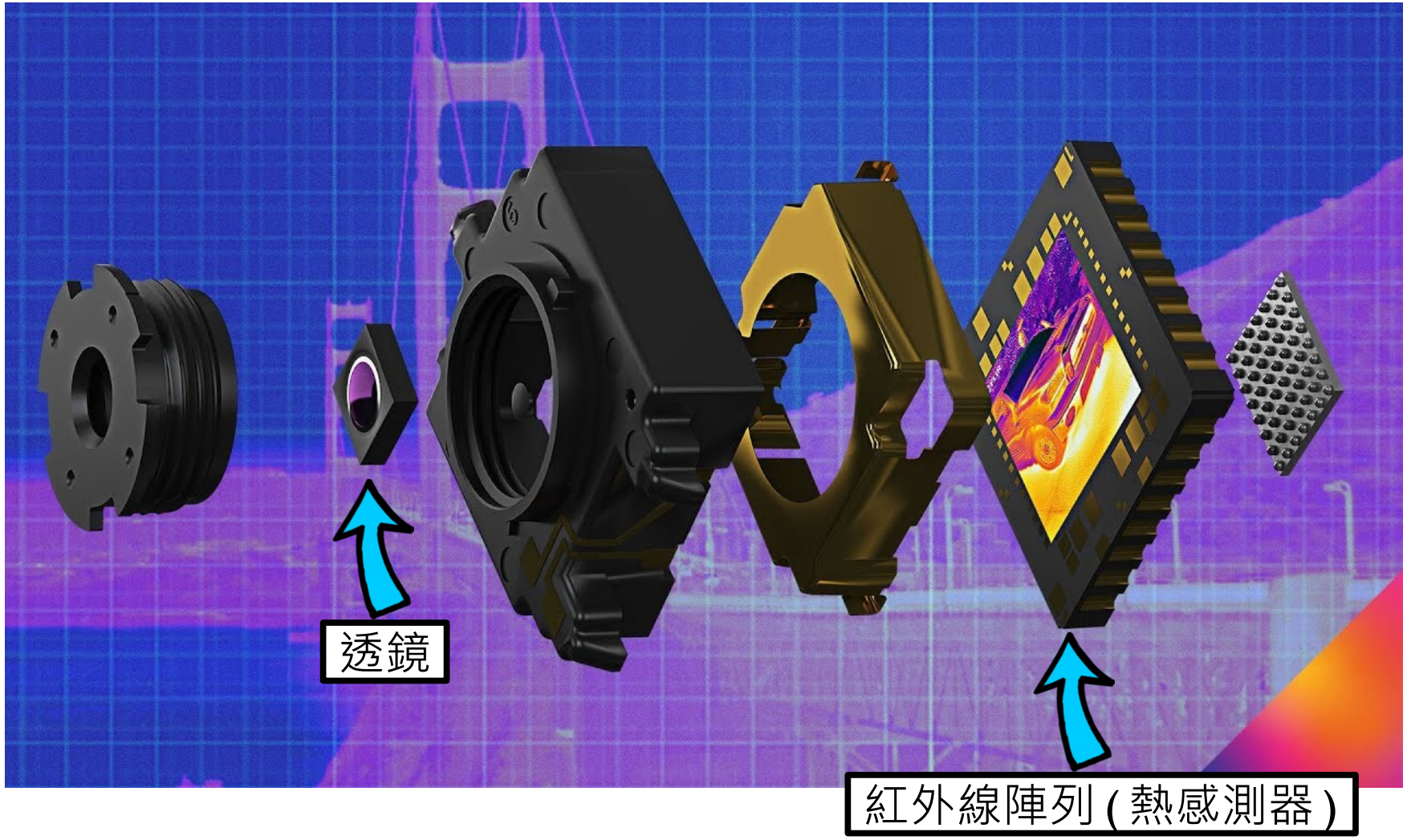
# 紅外線探測器 (Infrared detector) 類型

- 光子探測器 (Photonic)
  - 基於光電效應計算被吸收的光子數得到輻射分佈
  - 靈敏度高，使用波段窄，但需要冷卻
- 熱探測器 (Thermal)
  - 基於熱敏材料升溫後的電阻變化計算輻射能量
  - 需達到一定的輻射量才能計算溫度，累積計算
  - 非制冷型，可室溫下工作



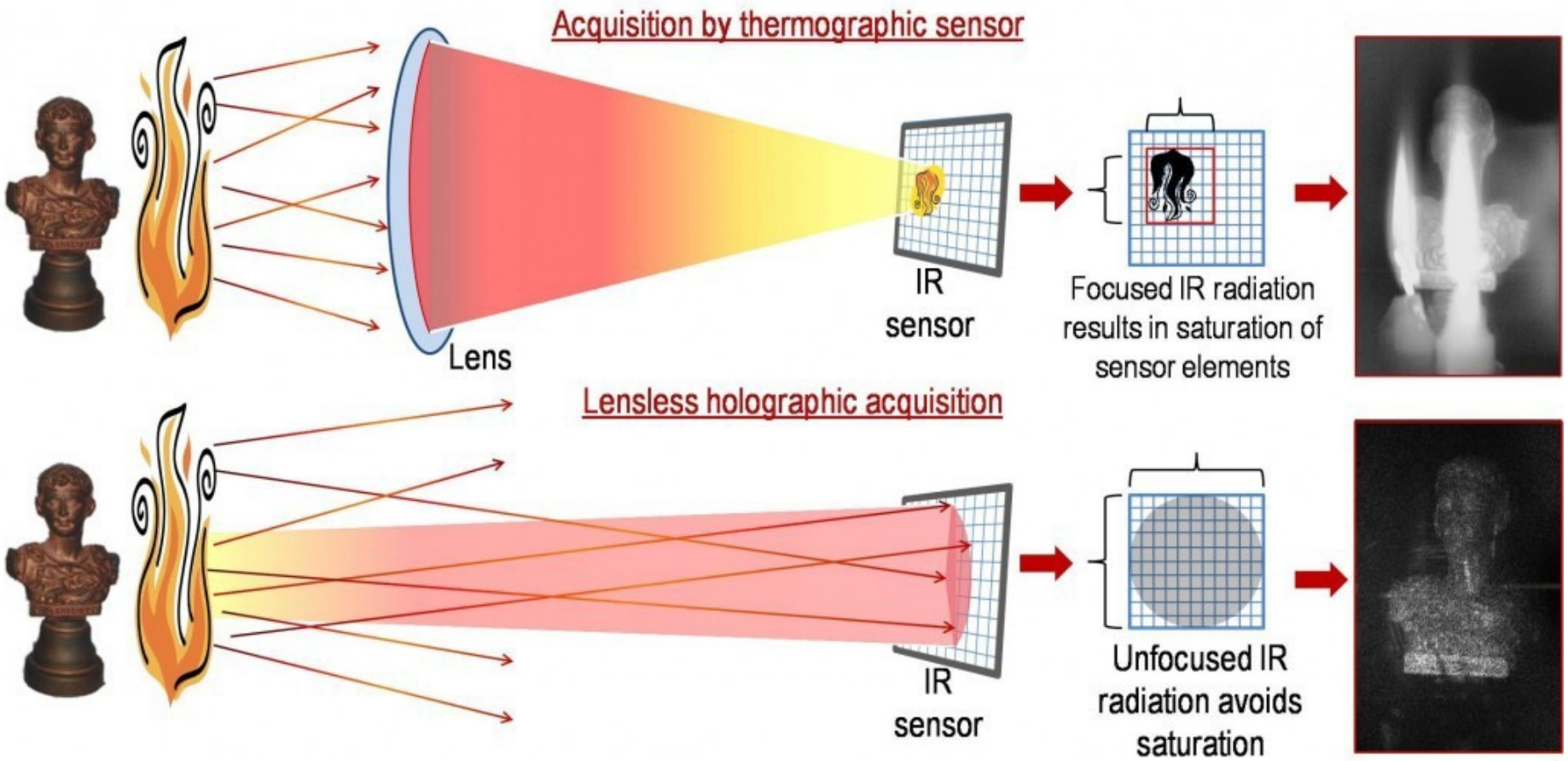

https://flir-scn.custhelp.com/app/answers/detail/a_id/1533

# 熱相機組成



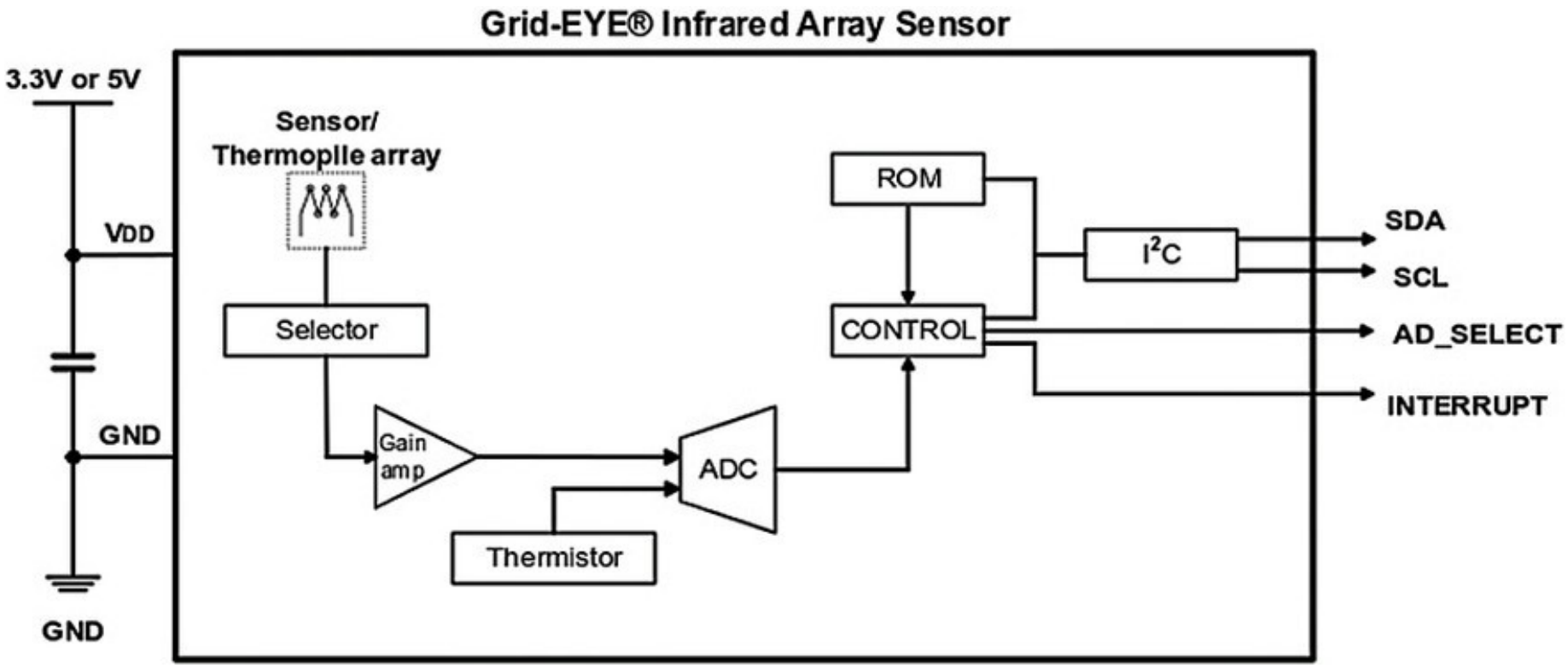

https://www.flir.com/discover/rd-science/how-do-thermal-cameras-work/

# 透鏡



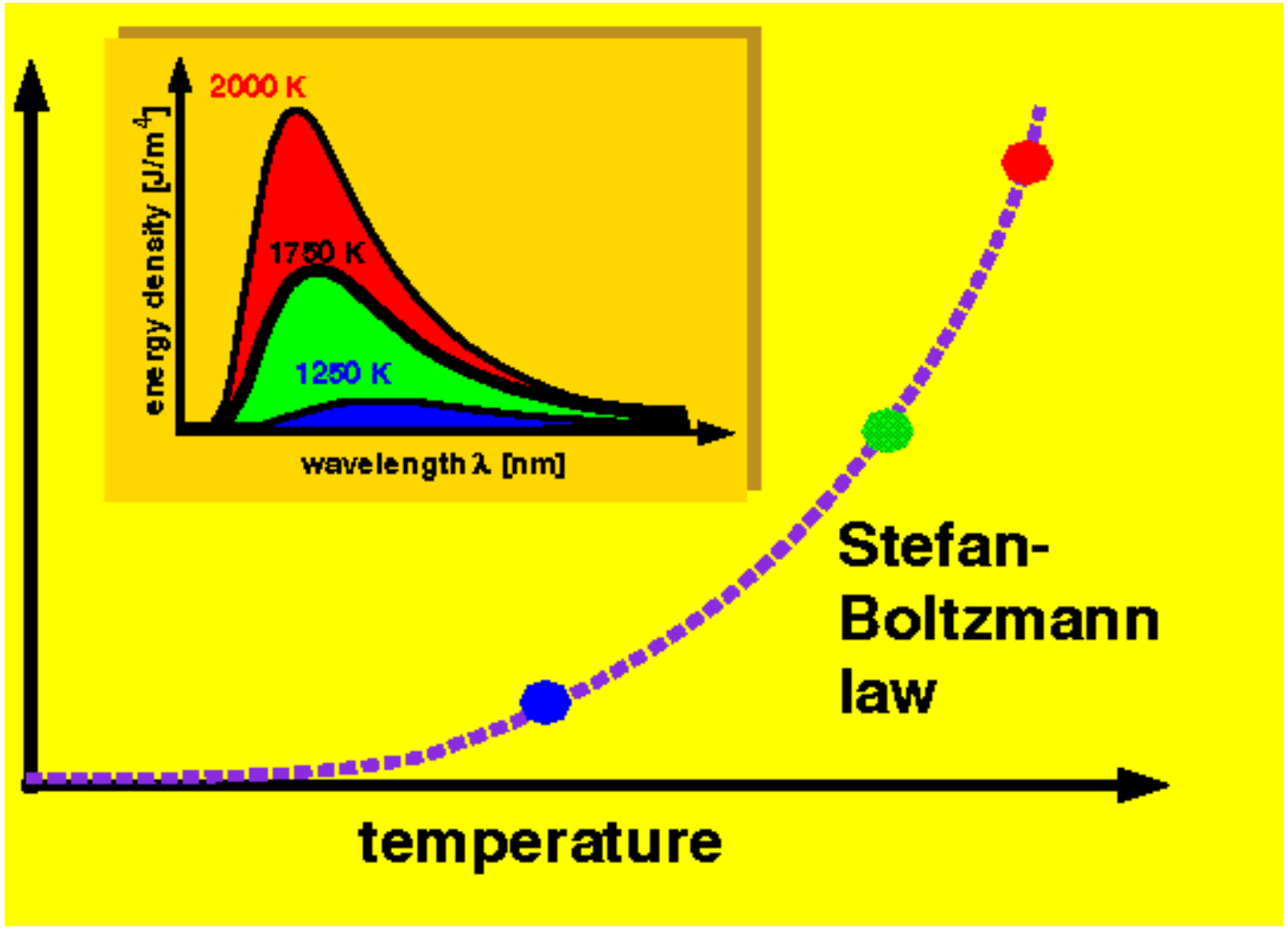

http://lumistarcorp.com/blog/?p=349

# 紅外線陣列感測溫度原理




https://www.fierceelectronics.com/components/how-to-build-eight-bit-thermal-imaging-camera

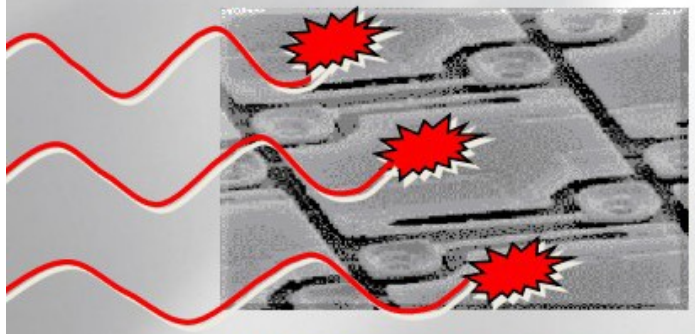
# 根據 Stefan-Boltzmann law 計算溫度




https://engineerjau.wordpress.com/tag/stefan-boltzmann-law/

# 熱影像顯示原理

## How IR Cameras Work



Photons Impact and
HEAT the Detector Element.

Stefan - Boltzmann Law

$$E = \varepsilon \sigma T^4$$

*Advanced Calculations*

Temperature Values are calculated

| | | | | | |
|---|---|---|---|---|---|
| 22 | 39 | 44 | 96 | 98 | 77 |
| 23 | 13 | 35 | 55 | 78 | 75 |
| 77 | 65 | 45 | 44 | 34 | 25 |
| 56 | 76 | 90 | 12 | 34 | 67 |
| 85 | 45 | 77 | 89 | 44 | 23 |
| 25 | 48 | 13 | 46 | 75 | 89 |

Colorized based on Temp Value



https://www.slideshare.net/flirsystems/flir-thermal-imaging-seminars-november-2015

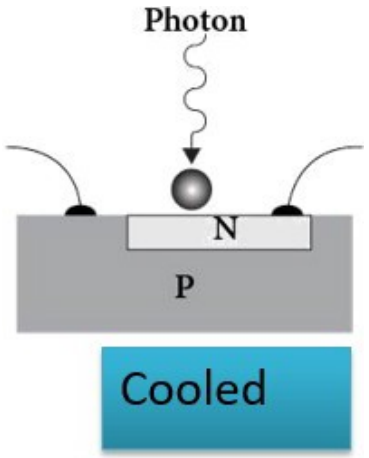
# 規格解讀

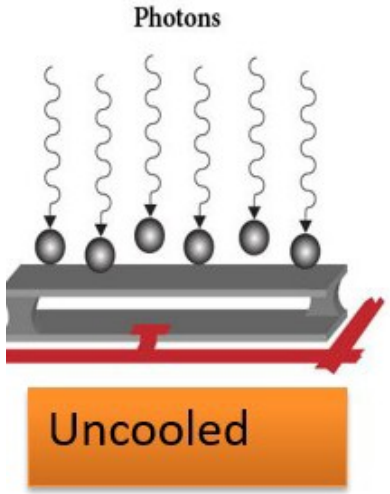
- 冷卻 (Cooled) / 非冷卻 (Uncooled)
  - 感測器不同物理性質的檢測方式
- 視場 (FOV, Field of View) / 瞬時視場 (IFOV, Instant FOV)
  - 相機可以看到的場景範圍
- 紅外線解析度 (IR Resolution)
  - 相機的像素數
- 熱敏度 (NETD, Noise Equivalent Temperature Difference)
  - 熱敏性或等效噪聲溫差 (NETD) 指相機可以看到的最小溫差
- 光譜範圍 (Spectral Range)
  - 指感測器可檢測到的波長範圍，以微米 (μm) 為單位

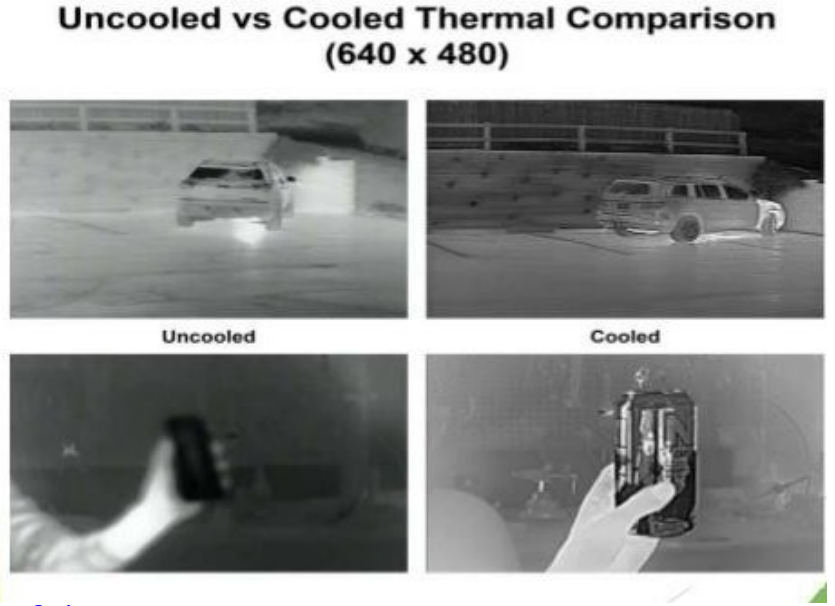
https://www.flir.com/discover/professional-tools/thermal-camera-specs-you-should-know-before-buying/

# Cooled/Uncooled Detector

- Cooled detector 工作溫度約為 -200°C, 高度敏感 (<20 mK), 測量中波紅外線 (3μ-5μ) 波長 (MWIR), 可中遠程檢測
- Uncooled detector 使用在室溫，敏感度為 300-200 mK, 長波紅外線 (LWIR) 波段 (7μ-14μ), 可在灰塵 / 霧氣和煙霧 等惡劣環境條件下有更好的穿透力



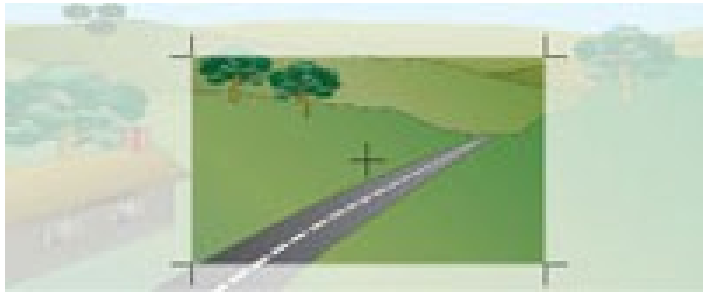



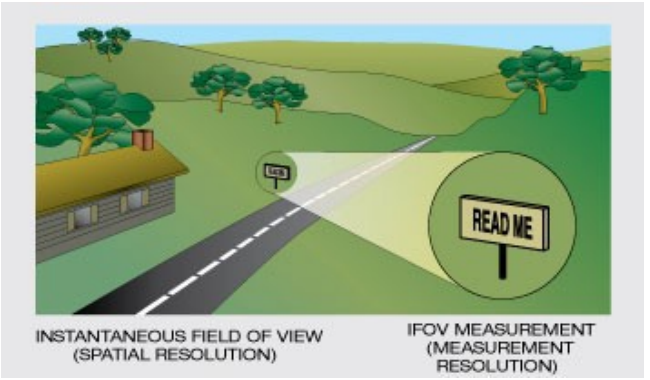

http://bit.ly/3fRs04x

# FOV / IFOV / IFOV Measurement

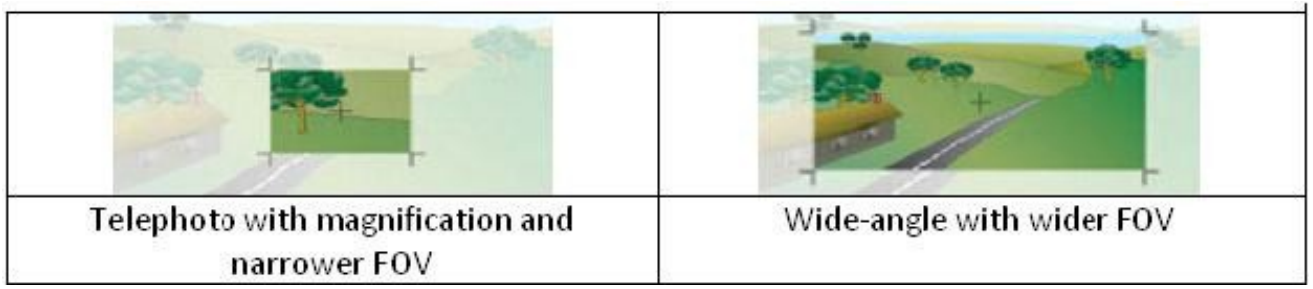
- FOV 是在設定距離內可以看到的最大區域
- IFOV 是 FOV 中，在設定距離可看到的最小細節
- IFOV Measurement，在設定距離上進行精確溫度測量的最小細節



FOV=hxv




IFOV

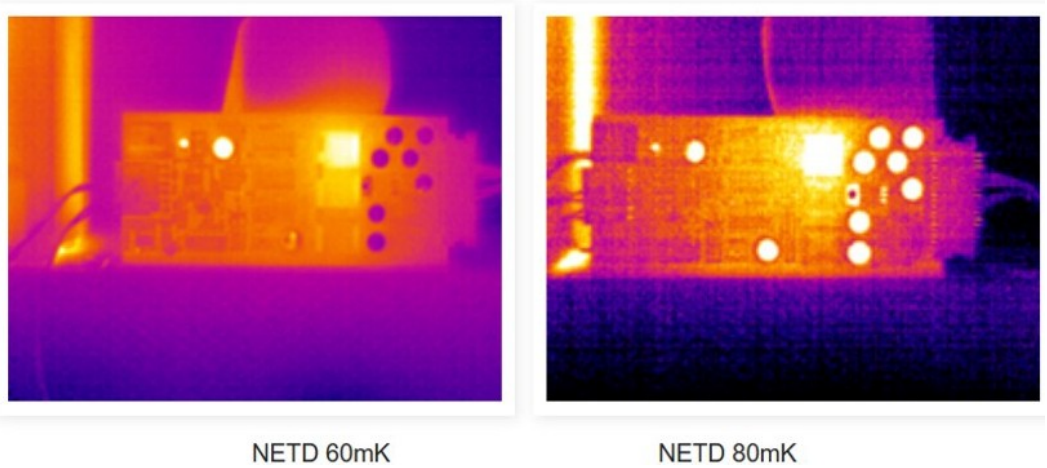



IFOV Measurement

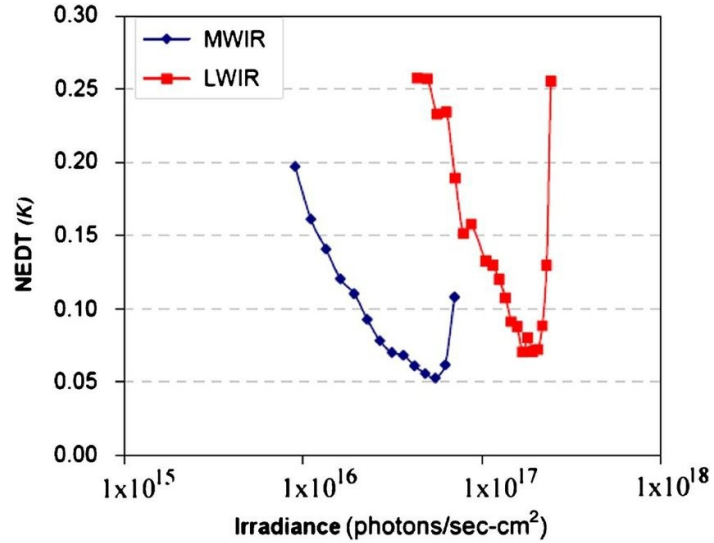
http://bit.ly/2OPnHLb

# NETD( 熱敏度 )

- NETD 代表溫差的信噪比數值，表示熱像儀可以分辨的最小溫差
- 計算公式為，瞬時噪聲除以響應度，單位 mK




不同 NETD 的影像結果




不同波長的 NETD

https://flir-scn.custhelp.com/app/answers/detail/a_id/558/related/1

# 影響熱相機準確度的原因

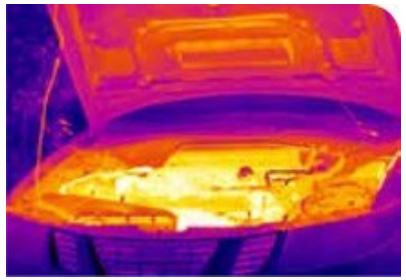
- 待測物體的發射率 (Emissivity)
- 環境溫度 (Ambient Temperature)
- 空間解析度 (Spatial Resolution)
- 光點尺寸 (Spot Size)
- 黑體與平均值 (Blackbody vs. Average)

https://www.convergint.com/thermal-camera-accuracy/

# 超級比一比(1)

# 點溫計 vs. 熱影像技術



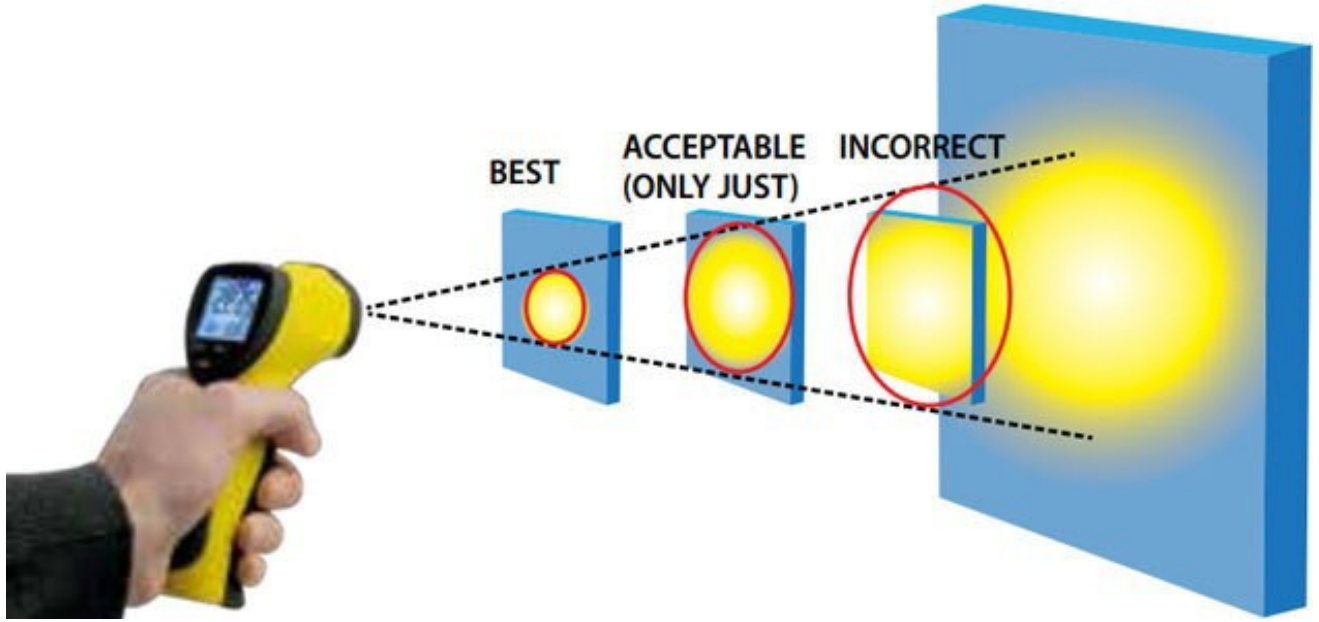


| | 點溫計（溫度槍） | 熱影像技術 |
|---|---|---|
| 別名 | Spot Pyrometer/ Infrared Thermometer | Thermal Imaging |
| 溫度讀取 | 單一點 | 全圖像 (>80x60) |
| 讀取距離 | 5-50cm | 5-700cm |

https://www.flir.com.au/discover/rd-science/temperature-guns-versus-thermal-imaging-technology

# 點溫計的溫度與距離考量

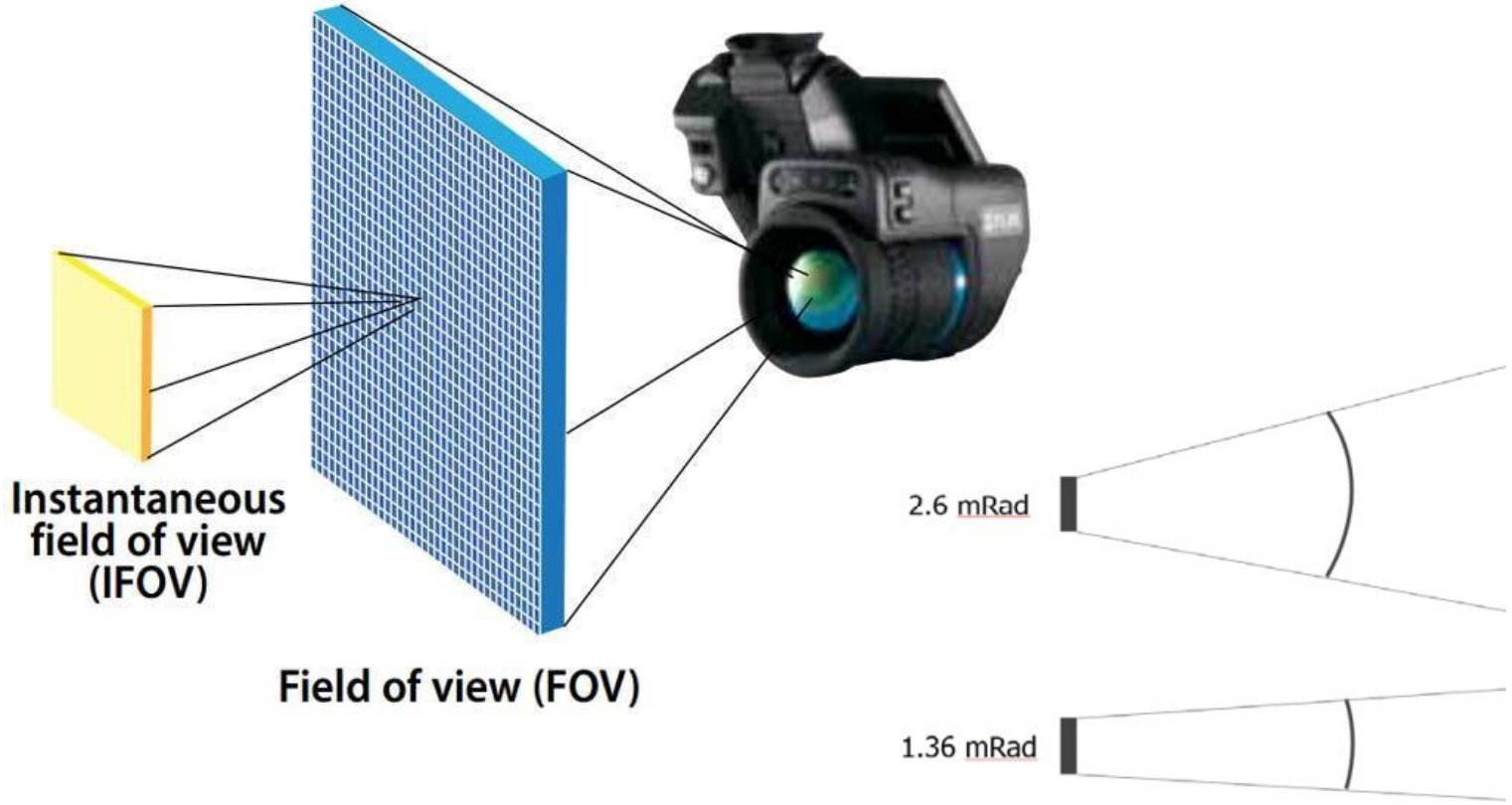
- 測量距離用點尺寸比 (SSR, Spot Size Ratio)
- 點溫計的 SSR 在 1:5 到 1:50(5-50cm 可測 1cm)




https://www.flir.com.au/discover/rd-science/temperature-guns-versus-thermal-imaging-technology

# 熱影像技術的溫度與距離考量

- 以瞬時視場(IFOV,Instantaneous Field of View)評估感測距離，單位毫弧度(mrad)




https://www.flir.com.au/discover/rd-science/temperature-guns-versus-thermal-imaging-technology

# IFOV 計算公式與 FLIR Lepton v3.5 實算結果

公式:

IFOV = (H_FOV / H_resolution) x (pi / 180) x 1000

H_FOV = 水平視場角(角度)，H_resolution = 水平解析度

pi / 180 = 角度轉弧度，1000 = 單位弧度轉為毫弧度

將 FLIR Lepton v3.5 的數值帶入 IFOV 公式:

IFOV = (56 / 160) x (pi / 180) x 1000 = 6.11 mrad

# 透過 IFOV 計算 SSR

假設 IFOV 視角切線近似於弧度值

Spot Size:測量距離 = (IFOV / 1000)

以剛剛 IFOV = 6.11 計算得到的 SSR = 163.6

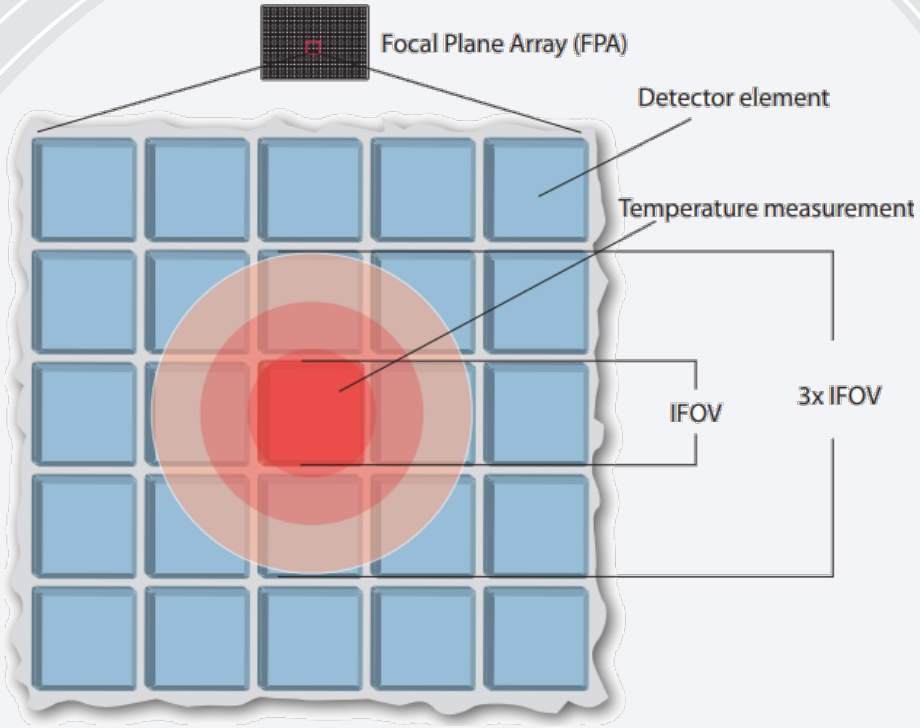
在實際的場景中使用的時候，我們會以測量視場MFOV替代IFOV

MFOV, Measurement Field of View = 3 x IFOV

因此實際使用的時候，計算得到的 SSR 為 54.6

即 IFOV = 6.11, SSR = 54.6

代表可以距離 54.6 cm 測量 1 cm 的物體

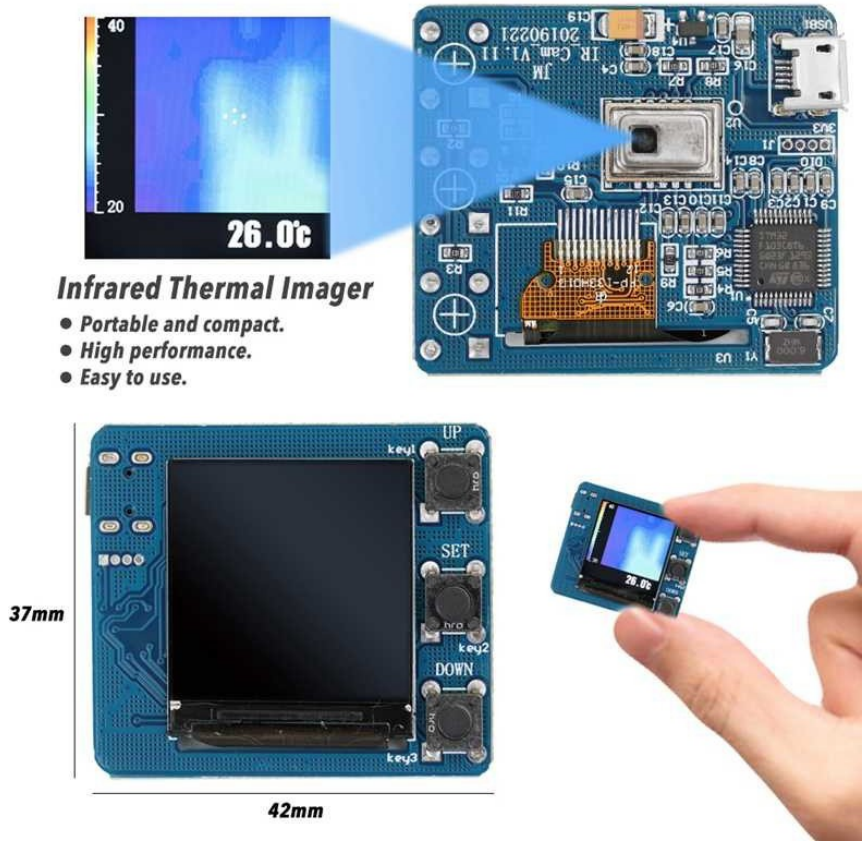


# 超級比一比 (2)

# 常用感測器比一比

- GY-906(MLX90614)
- AMG8833


GY-906 測單點



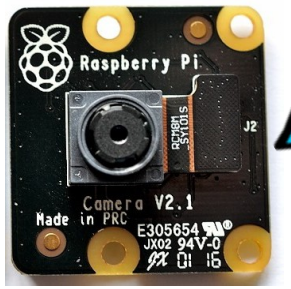

AMG8833 8x8 紅外線陣列

https://learn.sparkfun.com/tutorials/mlx90614-ir-thermometer-hookup-guide

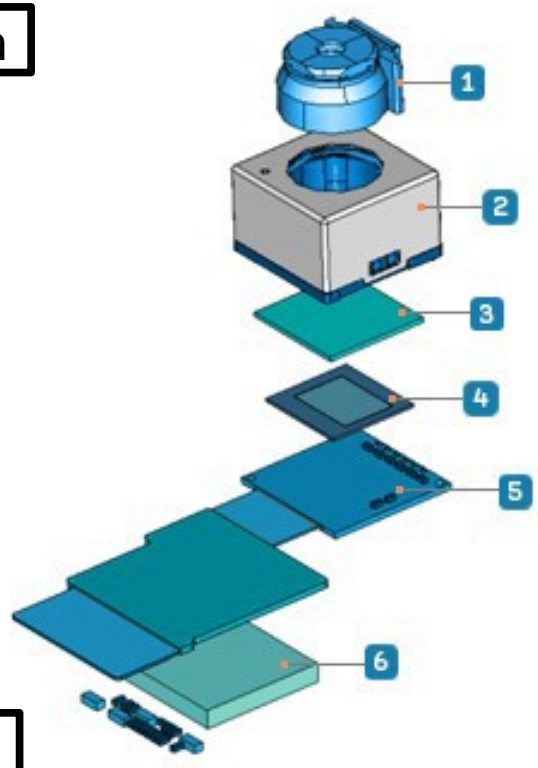
# 主動紅外線 vs. 熱影像技術

- 主動紅外線相機 (NoIR Camera) 需搭配短波紅外線 (NWIR) 發光源才能顯像

NoIR Camera




850nm 紅外光

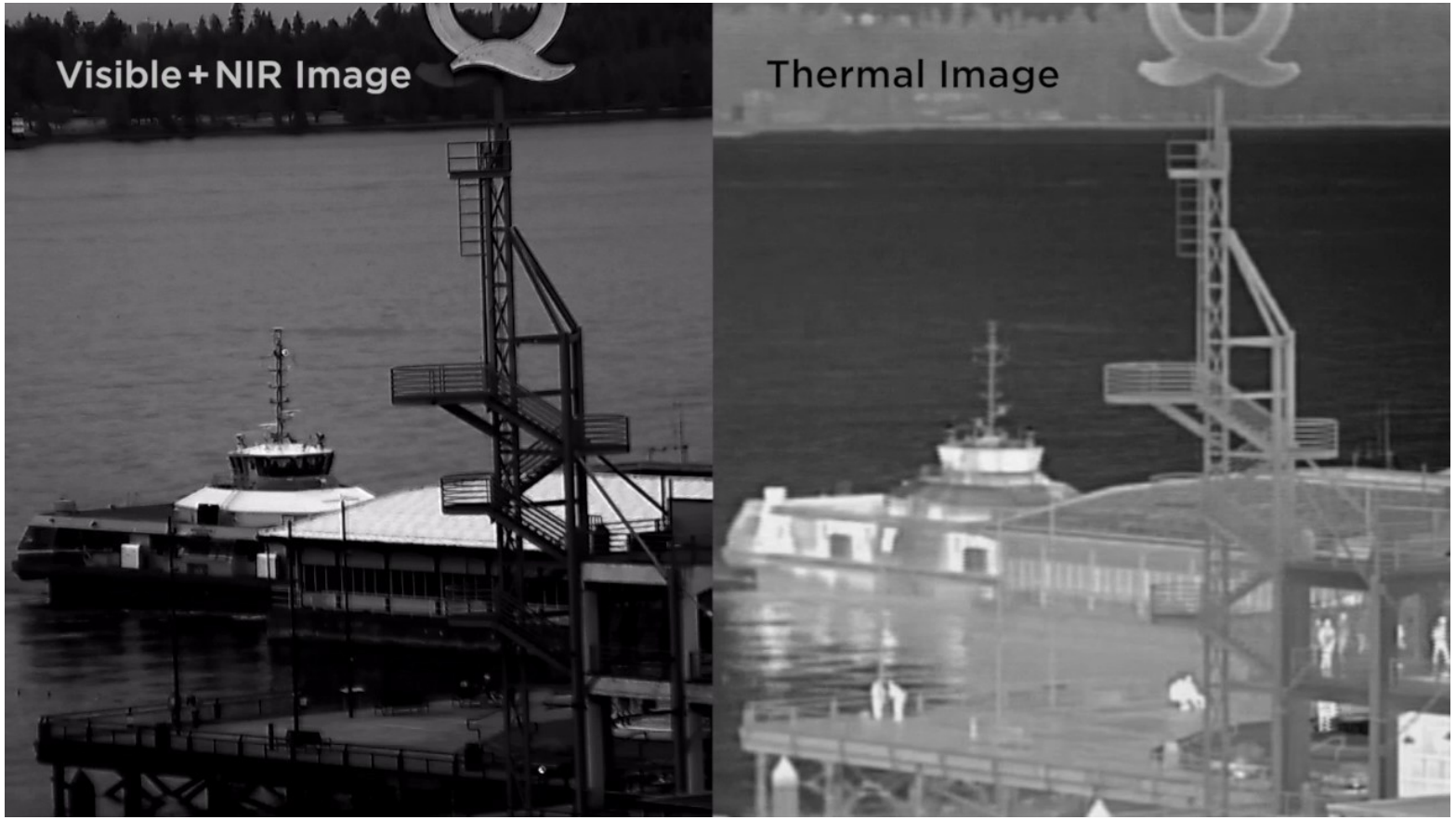



1. Lens( 透鏡 )
2. VCM( 音圈馬達 )
3. IR-Cut( 紅外光濾片 )
4. Sensor( 感光元件 )
5. PCB( 印刷電路板 )
6. ISP( 影像訊號處理器 )

http://www.phonearena.com/news/13MP-camera-tipped-for-Samsung-Galaxy-S-IV_id35168

# 熱影像技術屬於被動式顯像

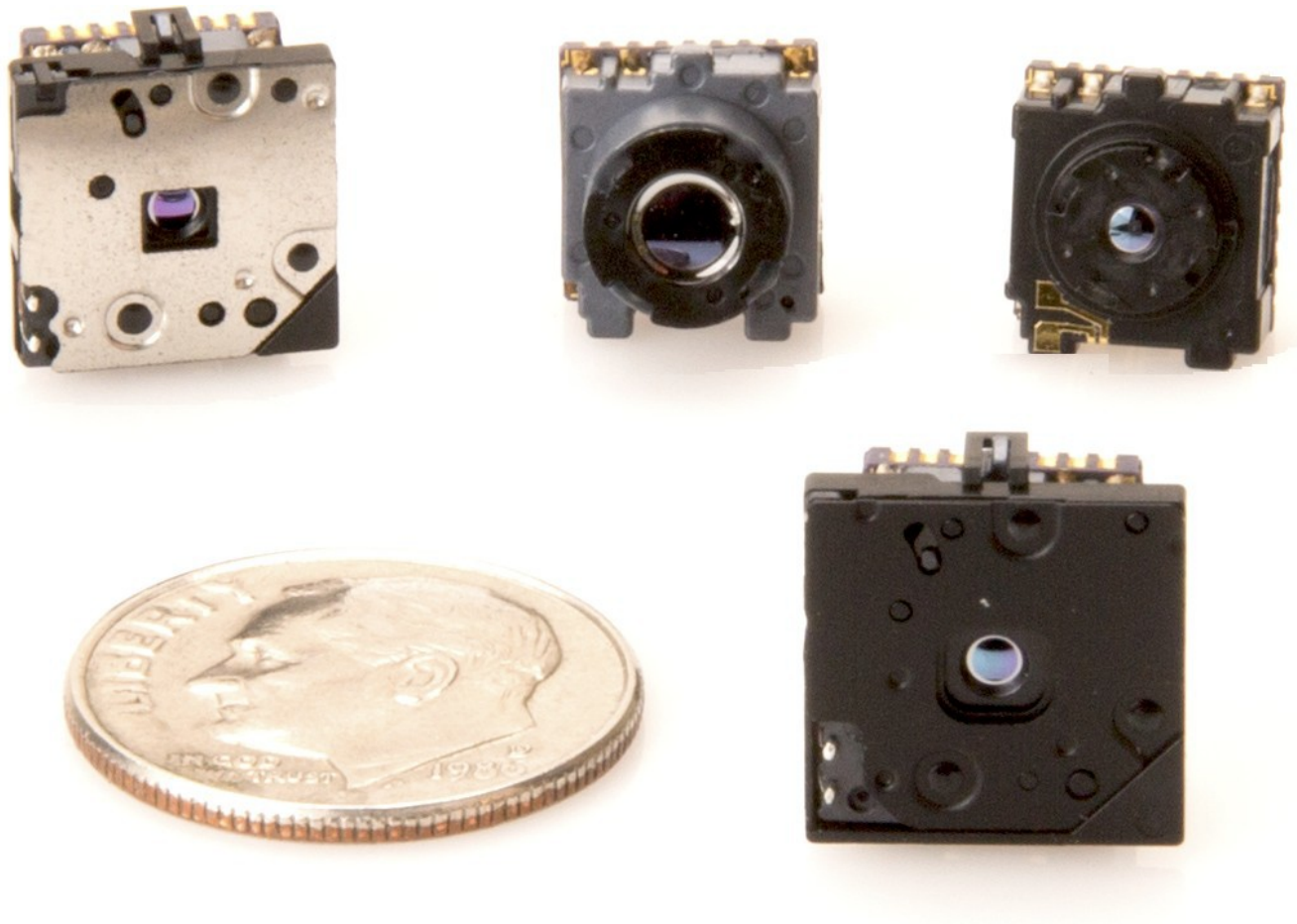
- 在完全無光的場景也能清楚顯示物體




https://www.youtube.com/watch?v=LLS_kvW81EU

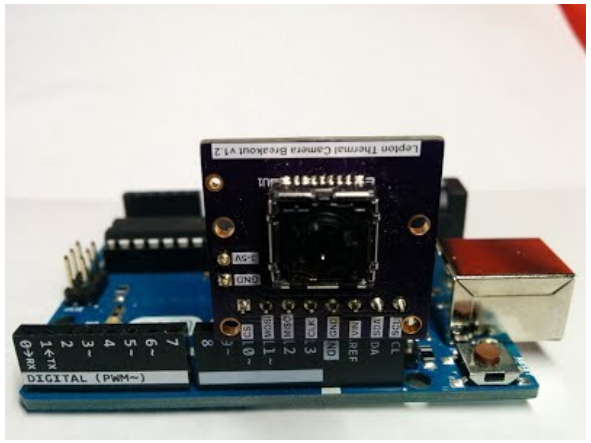
# 熱紅外線相機介紹

# Lepton LWIR Micro Thermal Camera Module

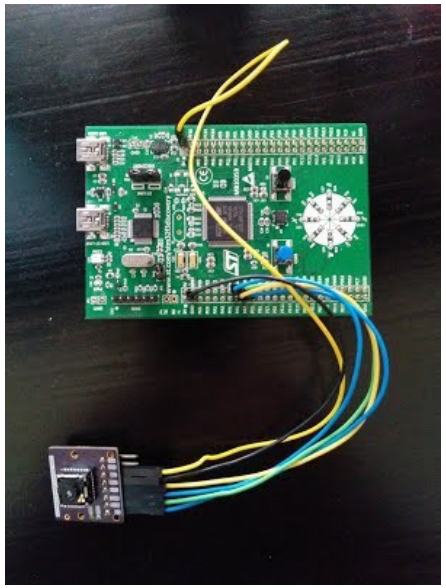


https://www.flir.com/products/lepton/

# 適用多種平台








http://www.pureengineering.com/projects/lepton

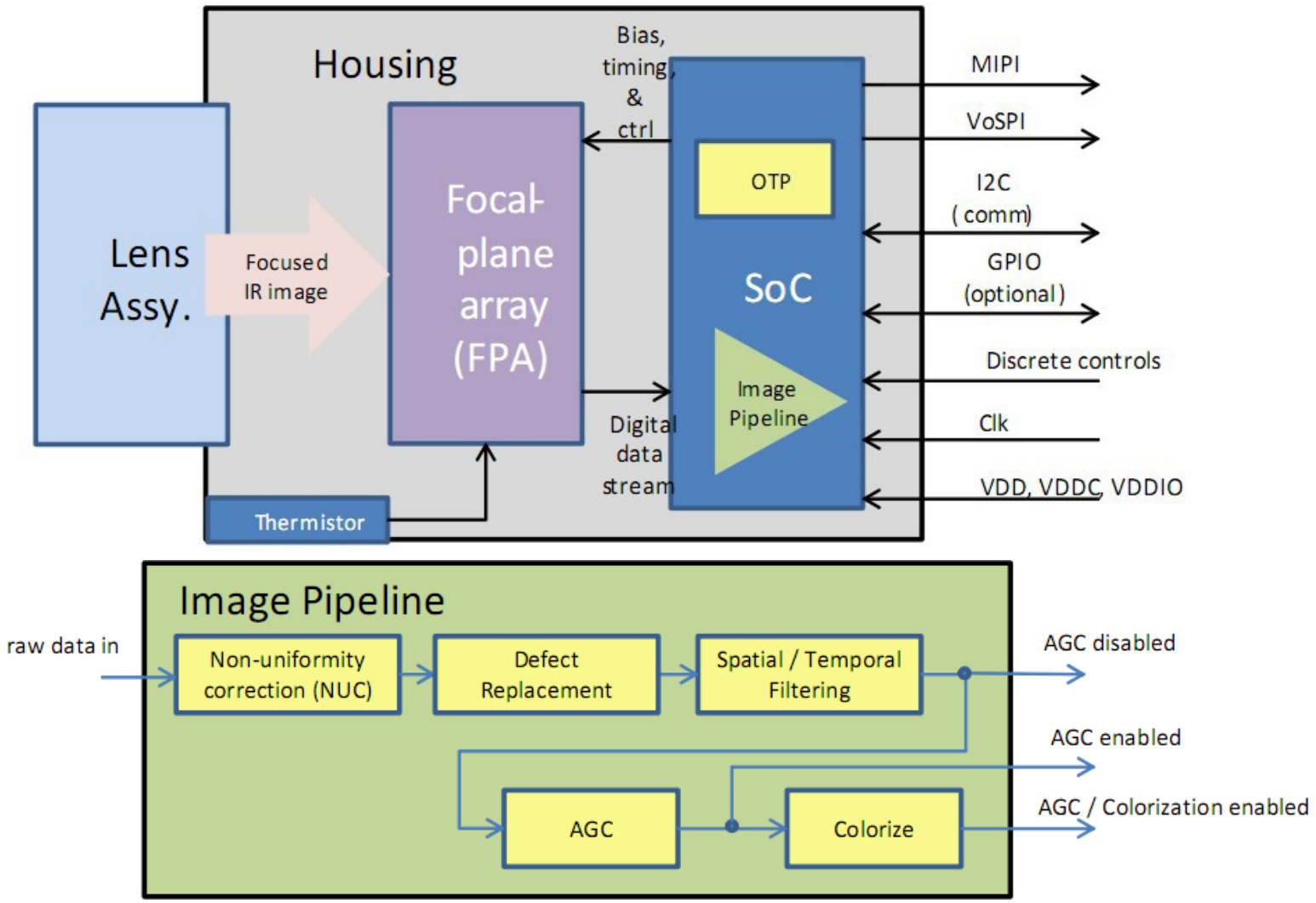
# FLIR Lepton 規格 比較

| | FLIR Lepton 2.0 | FLIR Lepton 2.5 | FLIR Lepton 3.5 |
|---|---|---|---|
| Resolution(H x V) | 80x60 | 80x60 | 160x120 |
| Spectral range | 8μm to 14 μm | 8μm to 14 μm | 8μm to 14 μm |
| Horizontal FOV | 51 | 50 | 56 |
| Diagonal FOV | | 63 | 71 |
| Thermal sensitivity | < 50 mK(0.05 C) | < 50 mK(0.05 C) | < 50 mK(0.05 C) |
| Frame rate (Hz) | 8.7 | 8.6 | 8.8 |
| Control interface | I2C | I2C | I2C |
| Video interface | SPI | SPI | SPI |
| Promised time to image | < 0.5 sec | < 1.2 sec | < 1.2 sec |
| Radiometric accuracy (35°C Blackbody) @ 25°C | | High gain: +/- 5°C or 5%.<br>Low gain: +/- 10°C or 10%. | High gain: ±5°C @ 25°C.<br>Low gain: ±10°C @ 25°C. |
| Radiometry | 14-bit pixel value | 14-bit pixel value, Kelvin | |
| Pixel size | 17um | 17um | 12um |

SSR=1:122 => 可以在 1.22m 測量 1cm 物體

https://flir.netx.net/file/asset/22586/original

# System Architecture + Video Pipeline




https://cdn.sparkfun.com/datasheets/Sensors/Infrared/FLIR_Lepton_Data_Brief.pdf

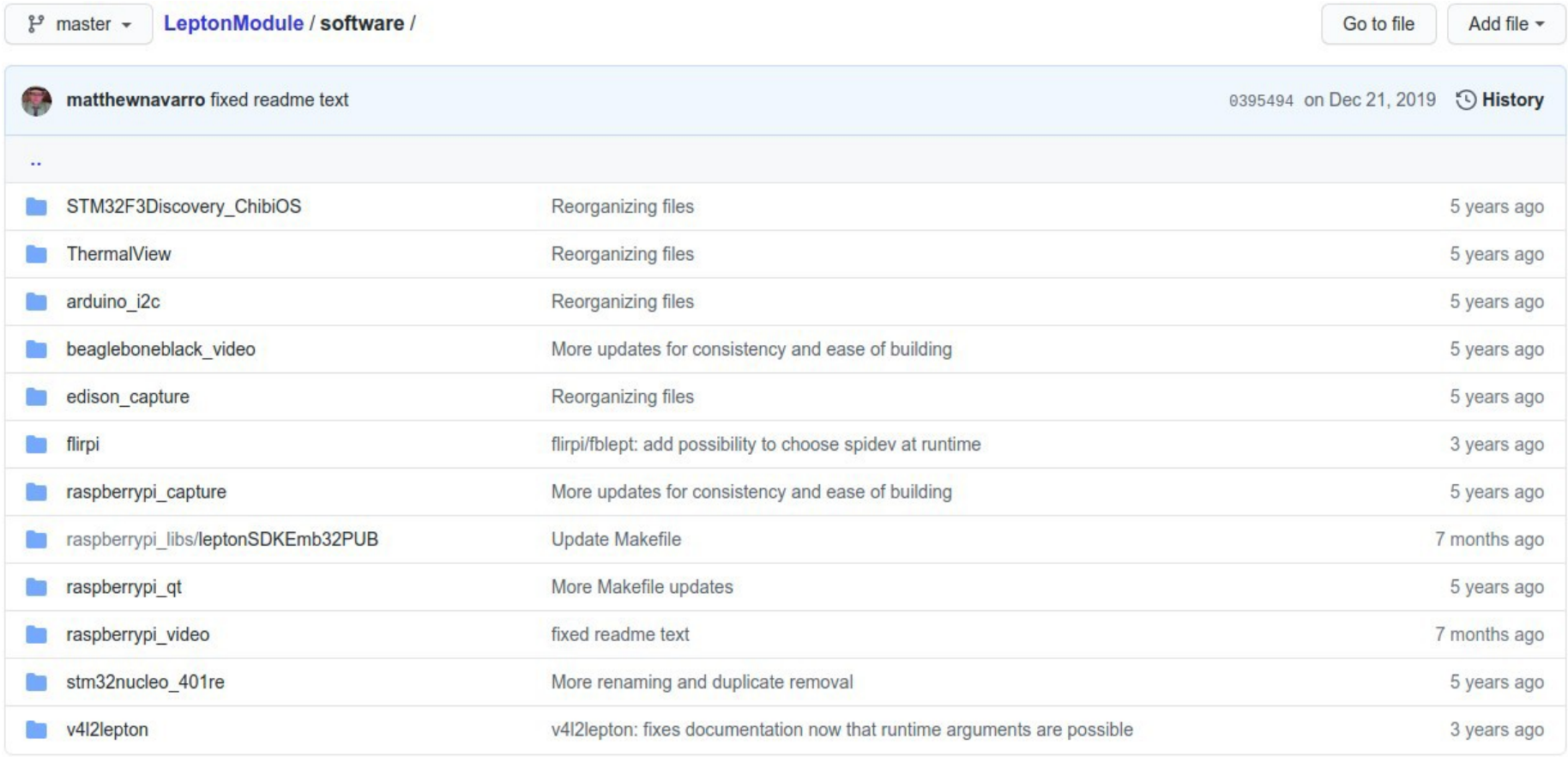
# Outline

A. 課前，必要軟硬體安裝
B. 原理講解
C. 實作測試
    1. 測試 FLIR 是否安裝正確
    2. 下載 LeptonModule
    3. 測試 拍照功能
    4. 測試 SPI 影片傳輸功能
D. OpenCV 安裝

# Lepton SDK

# Software Development Kit(SDK)

https://github.com/groupgets/LeptonModule



master ▾ LeptonModule / software / Go to file Add file ▾

matthewnavarro fixed readme text 0395494 on Dec 21, 2019 History

..

| | | |
|---|---|---|
| STM32F3Discovery_ChibiOS | Reorganizing files | 5 years ago |
| ThermalView | Reorganizing files | 5 years ago |
| arduino_i2c | Reorganizing files | 5 years ago |
| beagleboneblack_video | More updates for consistency and ease of building | 5 years ago |
| edison_capture | Reorganizing files | 5 years ago |
| flirpi | flirpi/fblept: add possibility to choose spidev at runtime | 3 years ago |
| raspberrypi_capture | More updates for consistency and ease of building | 5 years ago |
| raspberrypi_libs/leptonSDKEmb32PUB | Update Makefile | 7 months ago |
| raspberrypi_qt | More Makefile updates | 5 years ago |
| raspberrypi_video | fixed readme text | 7 months ago |
| stm32nucleo_401re | More renaming and duplicate removal | 5 years ago |
| v4l2lepton | v4l2lepton: fixes documentation now that runtime arguments are possible | 3 years ago |

https://github.com/groupgets/LeptonModule

# 下載測試 LeptonModule

- `$ cd ~`
- `$ git clone https://github.com/groupgets/LeptonModule`

- $ tree -L 2 LeptonModule
  - ├── docs/
  - ├── hardware/
  - └── software/
    - ├── arduino_i2c/
    - ├── beagleboneblack_video/
    - ├── edison_capture/
    - ├── flirpi/
    - ├── raspberrypi_capture/(測試拍照功能)
    - ├── raspberrypi_libs/(Lepton SDK)
    - ├── raspberrypi_qt/
    - ├── raspberrypi_video/(測試VoSPI功能)
    - ├── STM32F3Discovery_ChibiOS/
    - ├── stm32nucleo_401re/
    - ├── ThermalView/
    - └── v4l2lepton/(轉V4L2)

# 測試拍照功能
# (raspberrypi_capture)

# 拍照測試

先到檔案夾內編譯後執行

```
Bash - pi@thermal_rpi
$  cd ~/LeptonModule/software/raspberrypi_capture/
$  make
$  ./raspberrypi_capture
```

接下來應該會出現以下畫面

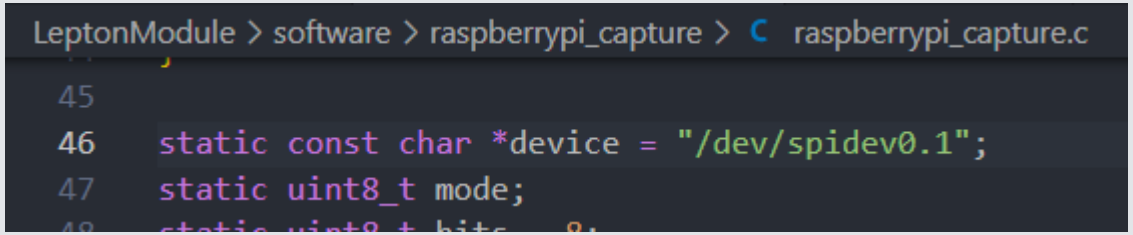
```
pi@thermalrpi:~/LeptonModule/software/raspberrypi_capture $ ./raspberrypi_capture
spi mode: 0
bits per word: 8
max speed: 16000000 Hz (16000 KHz)
^C
```

# 拍照測試

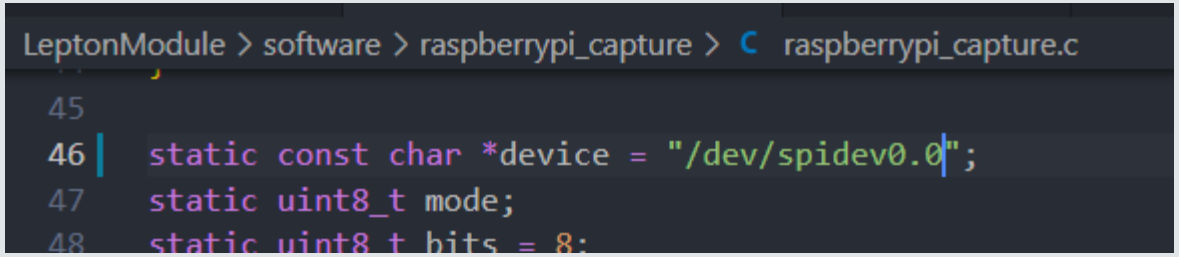
然後就會卡住，此時需要修改

`~/LeptonModule/software/raspberrypi_capture/raspberrypi_capture.c`

中的第 **46** 行

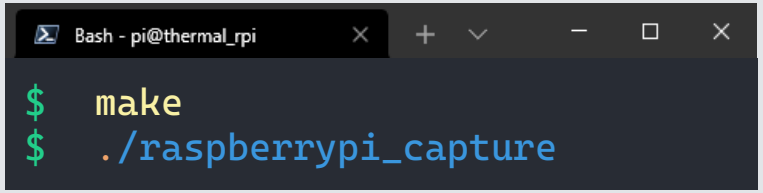



將 **spidev0.1** 改為 **spidev0.0**



# 拍照測試

修改完成後，重新編譯




接下來會出現以下輸出，正常的話**minval**與**maxval**都在**30k**上下

```
pi@thermalrpi:~/LeptonModule/software/raspberrypi_capture $ ./raspberrypi_capture
spi mode: 0
bits per word: 8
max speed: 16000000 Hz (16000 KHz)
Calculating min/max values for proper scaling...
maxval = 30396
minval = 29835
```

此時執行 **ls** 指令，可以看到輸出 **IMG_數字.pgm**，不過圖片內容會是錯誤的

```
pi@thermalrpi:~/LeptonModule/software/raspberrypi_capture $ ls
IMG_0000.pgm  raspberrypi_capture    raspberrypi_capture.o
Makefile      raspberrypi_capture.c  README.md
```

# 如果無法正確拍照

- 先確認 Breakout Board 版本？
- 再檢查接線是否正確？
- 檢查 raspi-config 中的 SPI 和 I2C 是否開啟？
- 重開機再試試看

# PGM 格式

- PBM 格式用 ASCII 碼表示單色點陣圖 (1988 年)
- PBM 是單色，PGM 是灰度圖，PPM 使用 RGB 顏色
- 檔案包含格式，圖像尺寸和實際資料

| 檔案描述子 | 類型 | 編碼 |
|---|---|---|
| P1 | 點陣圖 | ASCII |
| P2 | 灰度圖 | ASCII |
| P3 | 像素圖 | ASCII |
| P4 | 點陣圖 | 二進位 |
| P5 | 灰度圖 | 二進位 |
| P6 | 像素圖 | 二進位 |

```
P1
# This is an example bitmap of the letter "J"
6 10
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
1 0 0 0 1 0
0 1 1 1 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

PBM

```
P2
6 6
255
0 0 0 150 0 0
0 0 0 150 0 0
0 0 0 150 0 0
0 150 0 150 0 0
0 150 150 150 0 0
0 0 0 0 0 0
```

PGM

https://zh.wikipedia.org/wiki/PBM 格式

# raspberrypi_capture.c

pi@raspberrypi: ~/LeptonModule/software/raspberrypi_capture

File Edit Tabs Help

```
 84          for(i=0;i<60;i++)
 85          {
 86                  for(j=0;j<80;j++)
 87                  {
 88                          if (lepton_image[i][j] > maxval) {
 89                                  maxval = lepton_image[i][j];
 90                          }
 91                          if (lepton_image[i][j] < minval) {
 92                                  minval = lepton_image[i][j];
 93                          }
 94                  }
 95          }
 96          printf("maxval = %u\n",maxval);
 97          printf("minval = %u\n",minval);
 98
 99          fprintf(f,"P2\n80 60\n%u\n",maxval-minval);
100          for(i=0;i<60;i++)
101          {
102                  for(j=0;j<80;j++)
103                  {
104                          fprintf(f,"%d ", lepton_image[i][j] - minval);
105                  }
106                  fprintf(f,"\n");
107          }
                                                    84,2-9          45%
```

計算最大值和最小值

產生 PGM 檔案

# 測試 VoSPI 影像傳輸功能 (raspberrypi_video)

# SPI測試

先到檔案夾內編譯後執行，這邊的qmake是用qt編譯輸出Makefile文件

Bash - pi@thermal_rpi

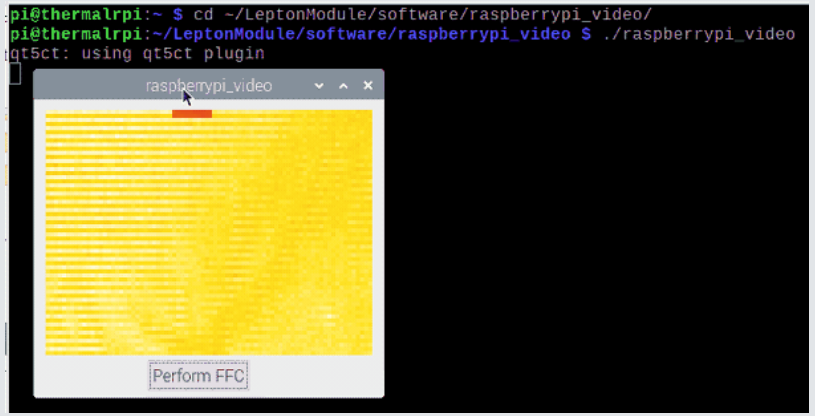
```
$  cd ~/LeptonModule/software/raspberrypi_video/
$  qmake -qt=qt5
$  make
$  ./raspberrypi_video
```

編譯完成後執行，應該會看到下面的左圖，

Bash - pi@thermal_rpi

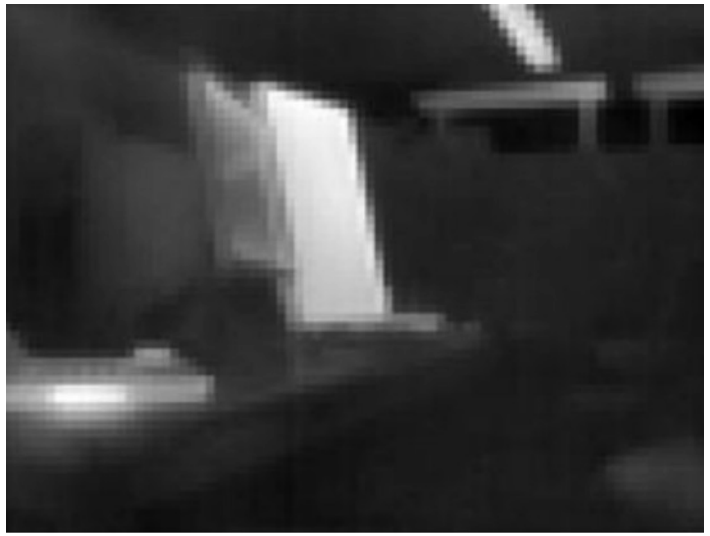
```
$  ./raspberrypi_video -tl 3
```

我們需要在執行後面增加參數 -tl 3 才會看到正常的右圖。

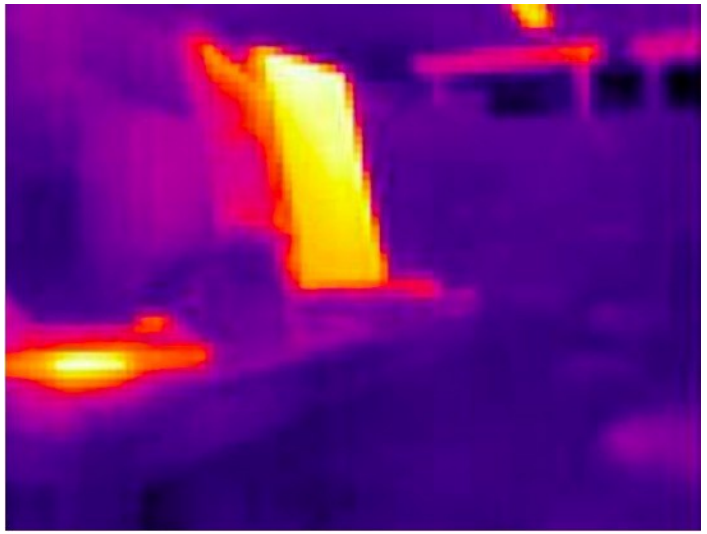



# Transferring Video over SPI

- 資料輸出格式
  - Raw Y14(一個14-bit灰階),原始光通量
  - RGB888(3個8-bit RGB),非真實顏色



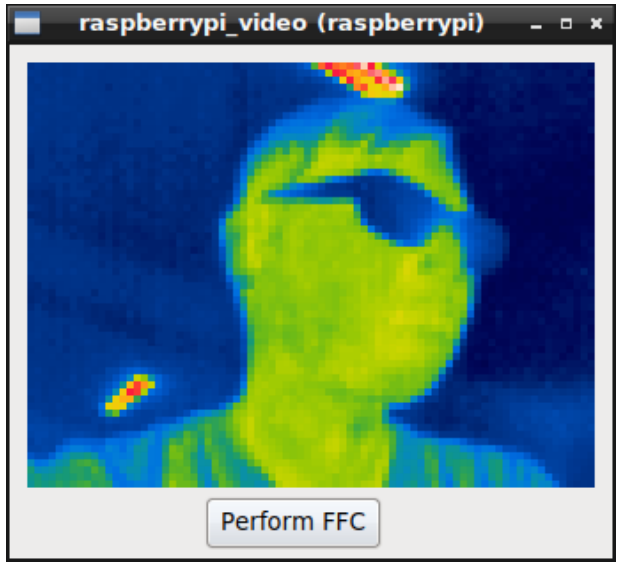
Raw Y14(Grayscale)



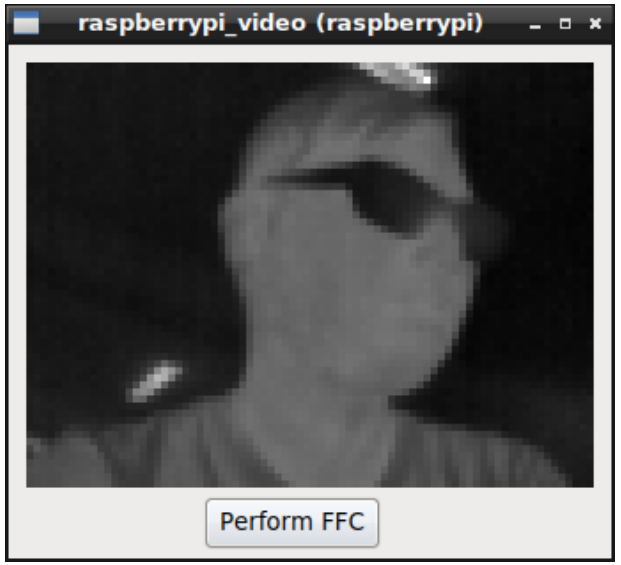
RGB888

# Color Map(Color Palette)

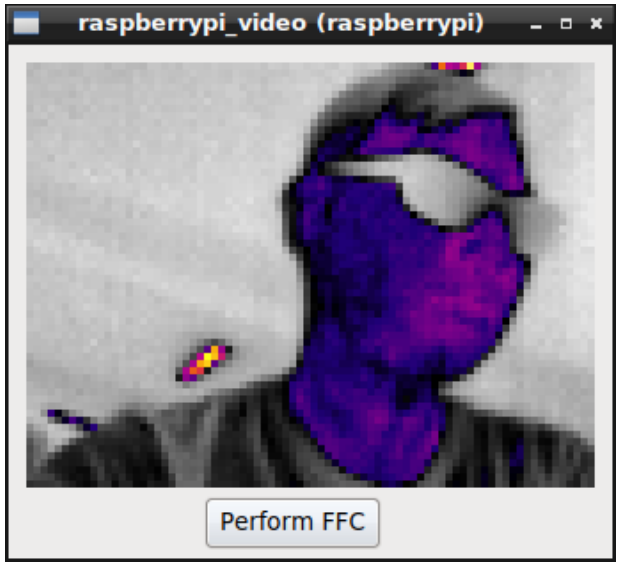
- 將熱影像可視化，常用的有




rainbow

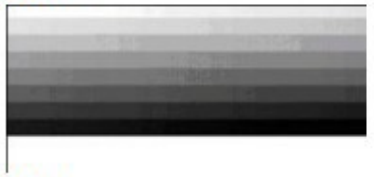
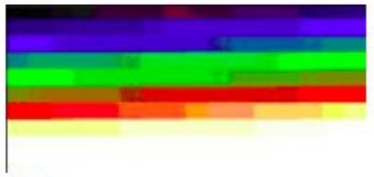
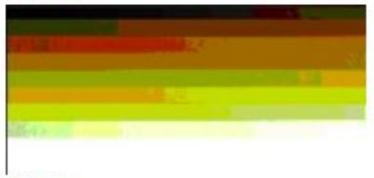
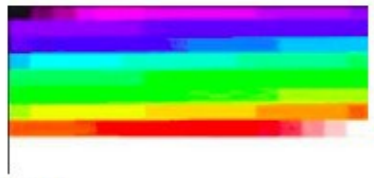
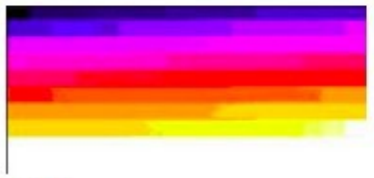



grayscale



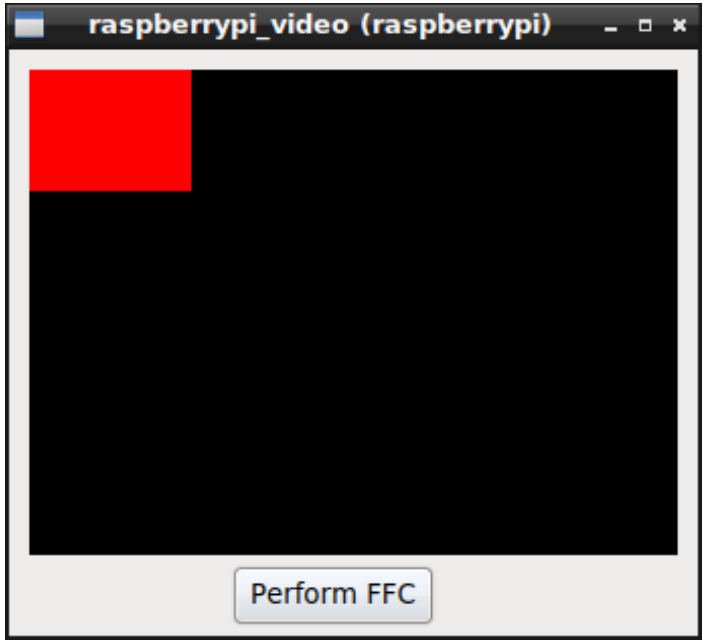

ironblack

# 更多 Color Map

| | | | |
|---|---|---|---|
| Cold  Hot **White Hot** | Cold  Hot **Black Hot** | Cold  Hot **Fusion** | Cold  Hot **Rainbow** |
| Cold  Hot **Glowbow** | Cold  Hot **Ironbow1** | Cold  Hot **Ironbow2** | Cold  Hot **Sepia** |
| Cold  Hot **Color1** | Cold  Hot **Color2** | Cold  Hot **Ice Fire** | Cold  Hot **Rain** |

http://flir-vue.de/wp-content/uploads/2015/07/flir_gui_user_guide.pdf

# 如果無法正確傳輸影像




- 先檢查接線是否正確？
- 檢查 `raspi-config` 中的 `SPI` 和 `I2C` 是否開啟？
- 可再執行一次 `raspberrypi_capture` 確認是否可拍照
- 重開機再試試看

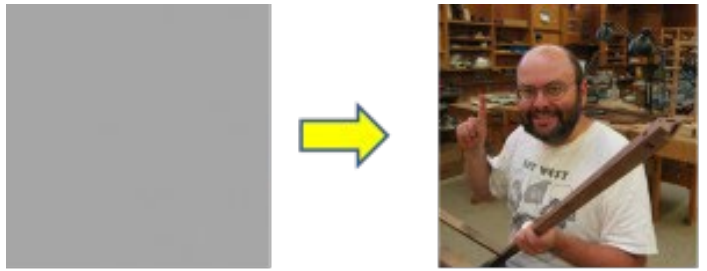
# Calibration

- 出廠校正 (Factory calibration)
  - 對每個相機的主要參數校正，不能在客戶端更改
- 平場校正 (FFC, Flat Field Correction)
  - 由於光照不均勻，鏡片中心和鏡片邊緣的反應不一致，或是成像器件各像元反應不一致，或是固定的影象背景噪聲等等
  - 平場校正就是通過改變每個像素響應直線的斜率 ( 訊號增益 Gain) 和偏移 ( 訊號偏移量 Offset)，使所有像素點的反應直線相同

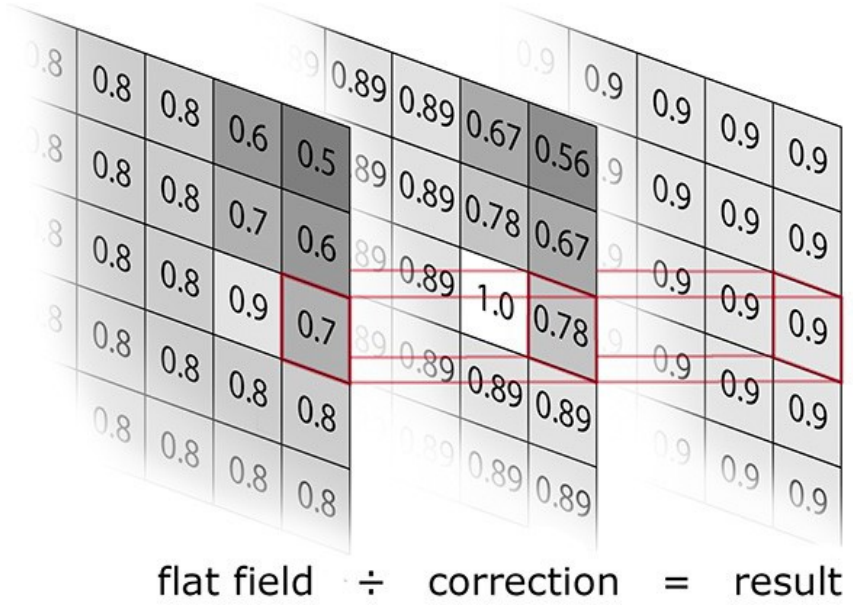
http://bit.ly/3fD3fcl

# Flat Field Correction(FFC)

均勻的光照 ( 上圖 )



不均勻的光照 ( 下圖 )




平場校正 (FFC) 目的

http://measurebiology.org/wiki/Flat-field_correction

# Outline

A. 課前，必要軟硬體安裝
B. 原理講解
C. 實作測試
D. OpenCV 安裝

# OpenCV 安裝

必要套件

# OpenCV 安裝

下載 OpenCV4.1 和 contrib

Bash - pi@thermal_rpi

```
$  cd ~
$  wget https://github.com/opencv/opencv_contrib/archive/4.1.1.tar.gz -O opencv_contrib-4.1.1.tar.gz
$  tar zxvf opencv_contrib-4.1.1.tar.gz
$  wget https://github.com/opencv/opencv/archive/4.1.1.tar.gz
$  tar -zxvf 4.1.1.tar.gz
$  cd opencv-4.1.1
$  mkdir build
$  cd build
```

# OpenCV 安裝

編譯安裝 OpenCV4.1

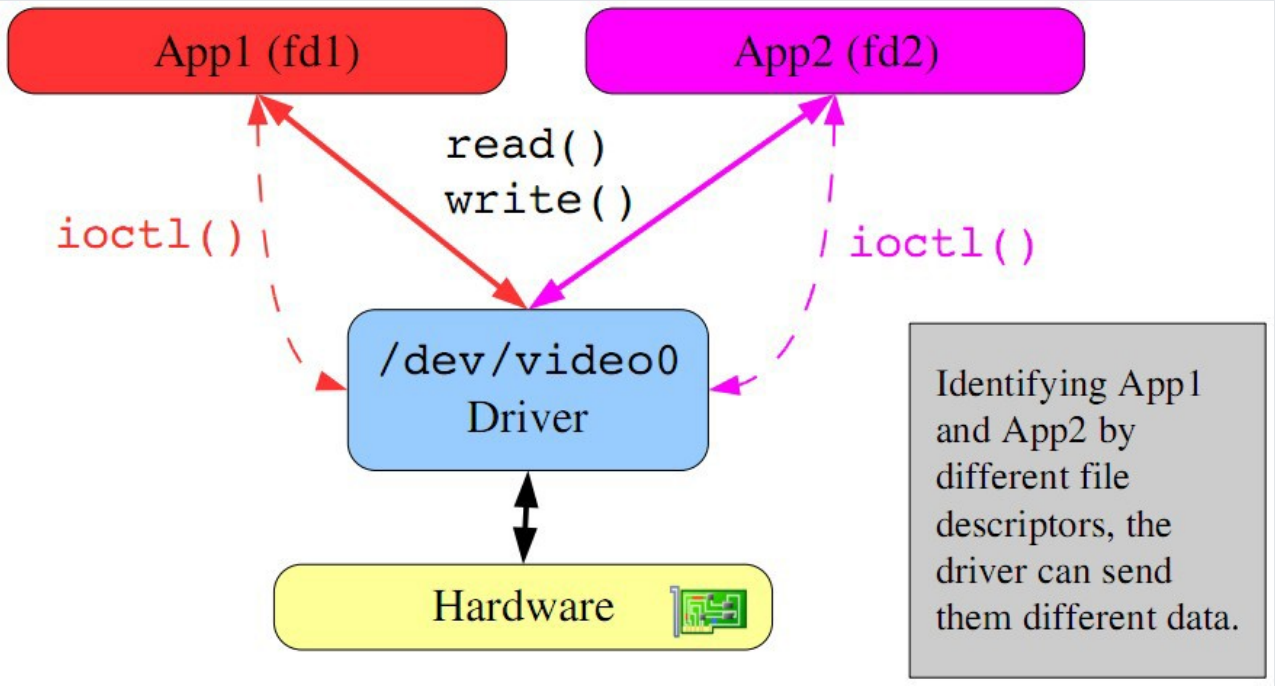
Bash - pi@thermal_rpi

```
$  cmake -D CMAKE_BUILD_TYPE=RELEASE \
$    -D CMAKE_INSTALL_PREFIX=/usr/local \
$    -D OPENCV_EXTRA_MODULES_PATH=/home/pi/opencv_contrib-4.1.1/modules \
$    -D ENABLE_NEON=ON \
$    -D ENABLE_VFP3=ON \
$    -D BUILD_TESTS=OFF \
$    -D INSTALL_C_EXAMPLES=ON \
$    -D INSTALL_PYTHON_EXAMPLES=ON \
$    -D BUILD_EXAMPLES=ON \
$    -D OPENCV_ENABLE_NONFREE=ON \
$    -D CMAKE_SHARED_LINKER_FLAGS=-latomic ..
$  time make -j4 VERBOSE=1
$  sudo make install
```

# 使用V4L2 讀取影像

# Video For Linux 2nd(V4L2)

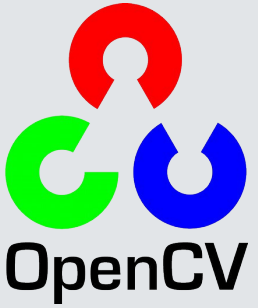
- 是 Linux 對視訊設備 ( 如 Webcam) 的 Userspace API



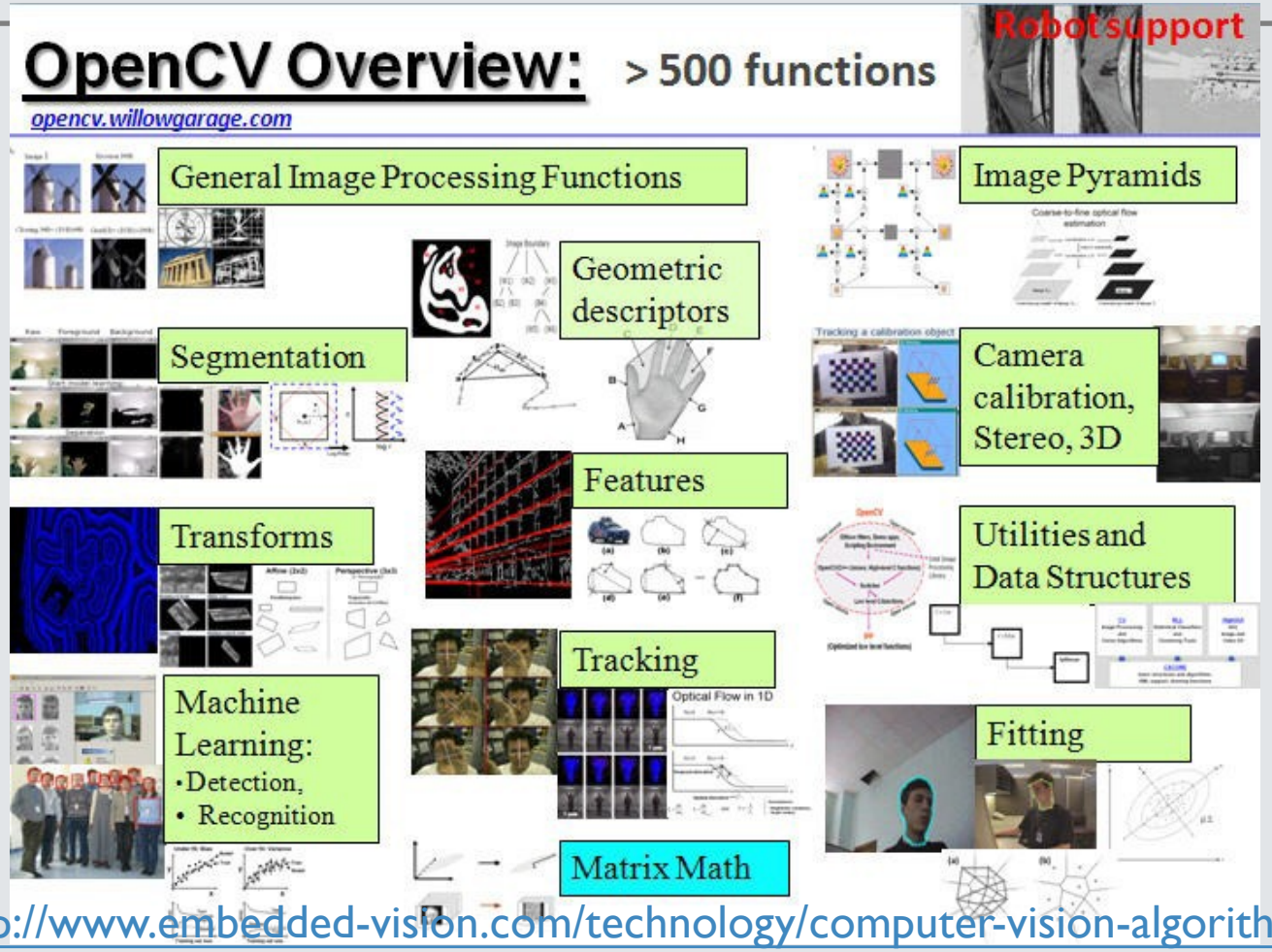

http://free-electrons.com/doc/embedded_linux_multimedia.pdf
https://www.linuxtv.org/downloads/legacy/video4linux/API/V4L2_API/spec-single/v4l2.html

# 使用OpenCV讀取，處理影像

# OpenCV
## - Open Source Computer Vision Library

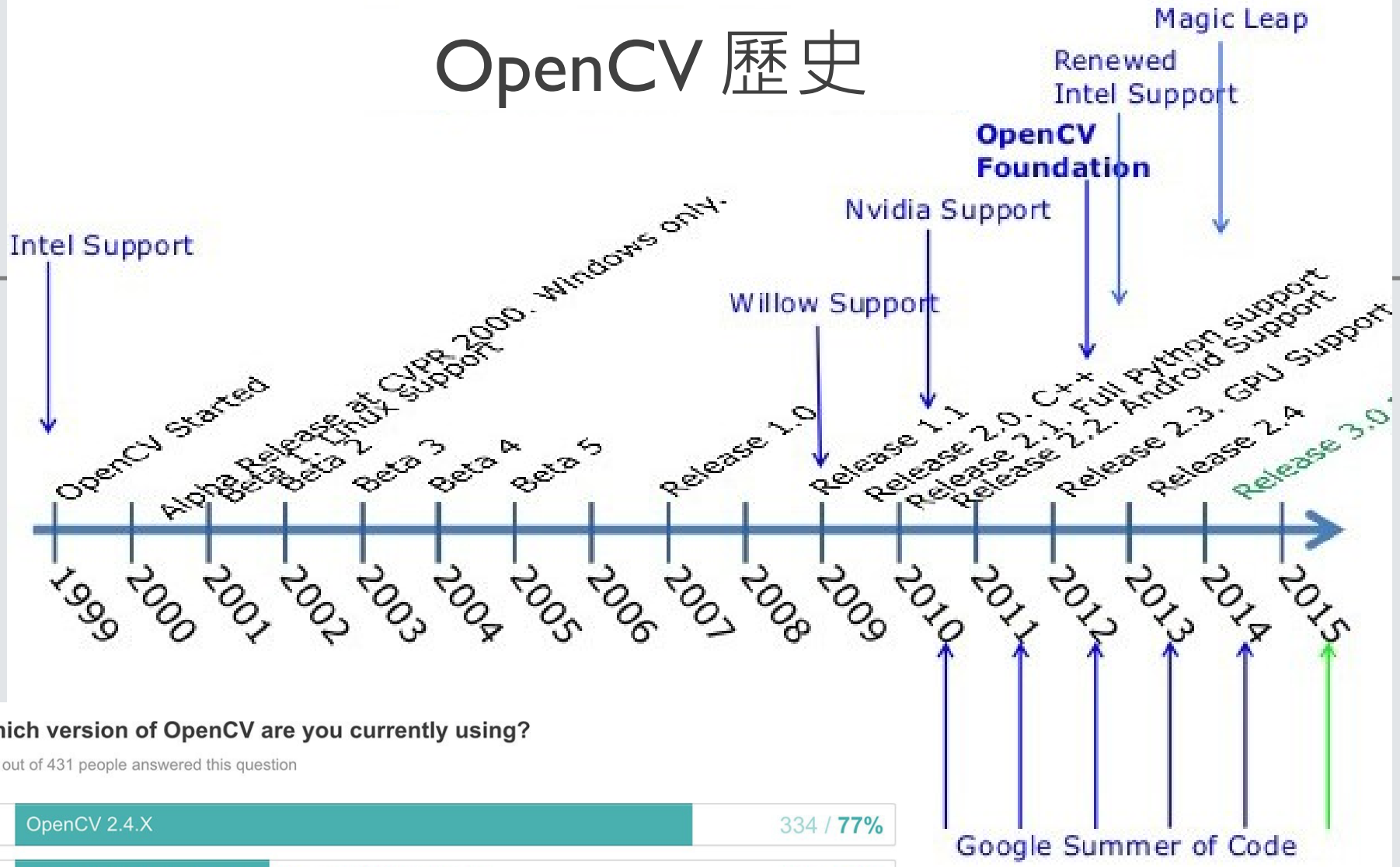



- 跨平台的計算機函式庫，主要由 C/C++ 撰寫




http://www.embedded-vision.com/technology/computer-vision-algorithms

# OpenCV 歷史




**Which version of OpenCV are you currently using?**

**431** out of 431 people answered this question

| | | |
|---|---|---|
| 1 | OpenCV 2.4.X | 334 / **77%** |
| 2 | OpenCV 3.0 (includes beta, RC, and official release) | 125 / **29%** |
| 3 | Other | 9 / **2%** |

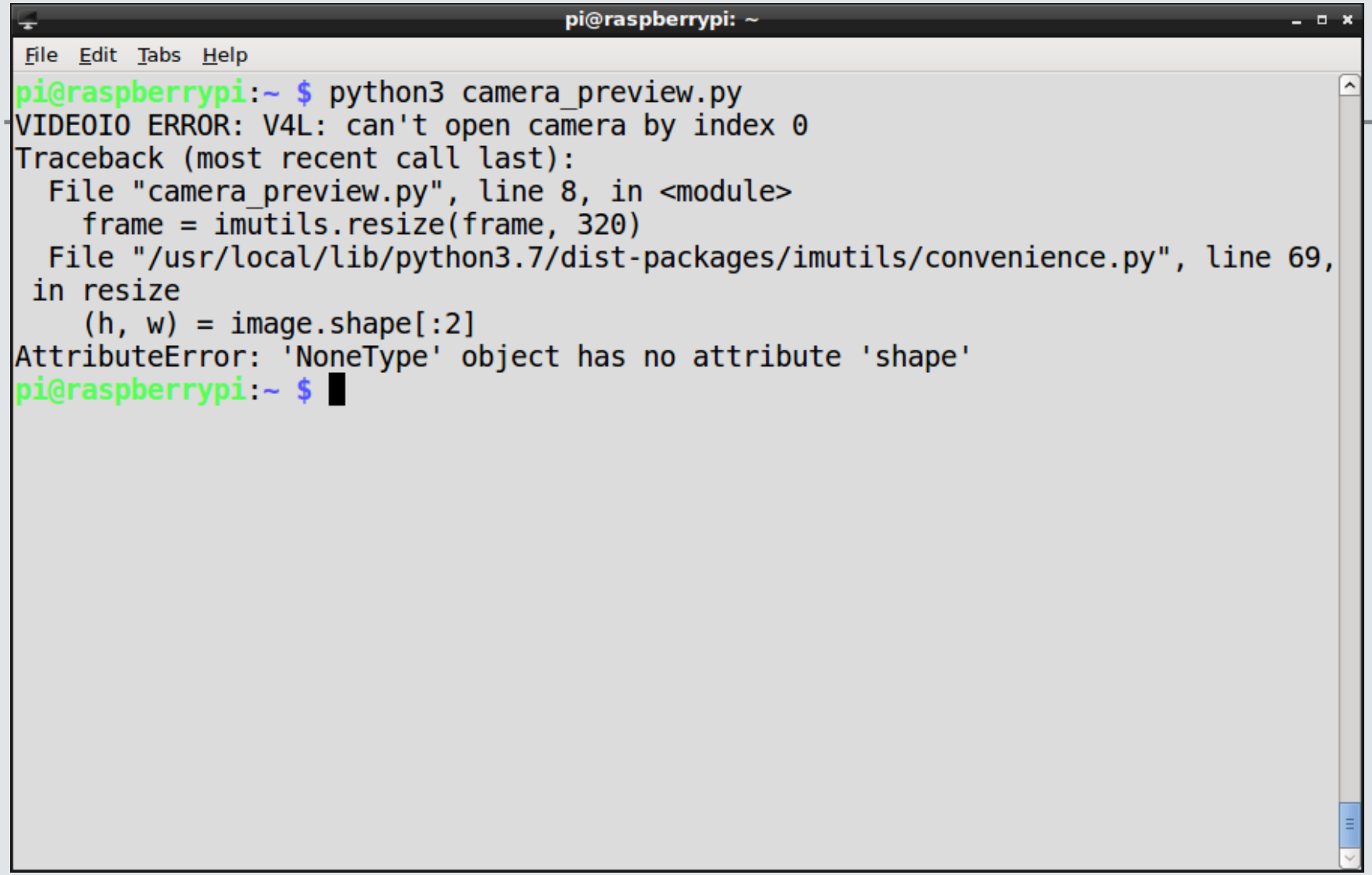
https://www.slideshare.net/embeddedvision/e04-open-cvbradsk

OpenCV

# 讀取Camera 並顯示

```
import cv2
import imutils

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    frame = imutils.resize(frame, 320)
    cv2.imshow("preview", frame)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

cap.release()
cv2.destroyAllWindows()
```

# 執行結果


pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ python3 camera_preview.py
VIDEOIO ERROR: V4L: can't open camera by index 0
Traceback (most recent call last):
File "camera_preview.py", line 8, in <module>
frame = imutils.resize(frame, 320)
File "/usr/local/lib/python3.7/dist-packages/imutils/convenience.py", line 69,
in resize
(h, w) = image.shape[:2]
AttributeError: 'NoneType' object has no attribute 'shape'
pi@raspberrypi:~ $

# 用V4L2 讀取 SPI 影像(v4l2loopback)

- 步驟：

1. 安裝編譯 v4l2loopback 必要軟體
2. 編譯 v4l2loopback 虛擬裝置節點
3. 安裝 V4L2 Kernel Module
4. 使用 V4L2 讀取 SPI 影像

https://github.com/raspberrypi-tw/meetup19

# 1.安裝必要軟體(已安裝)

- `$ sudo apt-get update`
- `$ sudo apt-get install -y bc flex bison libncurses5-dev`
- `$ sudo wget https://raw.githubusercontent.com/notro/rpi-source/master/rpi-source -O /usr/bin/rpi-source && sudo chmod +x /usr/bin/rpi-source && /usr/bin/rpi-source -q --tag-update`
- `$ rpi-source`

詢問問題時，都按 Enter( 預設答案 )

# 2. 編譯v4l2loopback 虛擬裝置節點

- `$ cd ~`

---

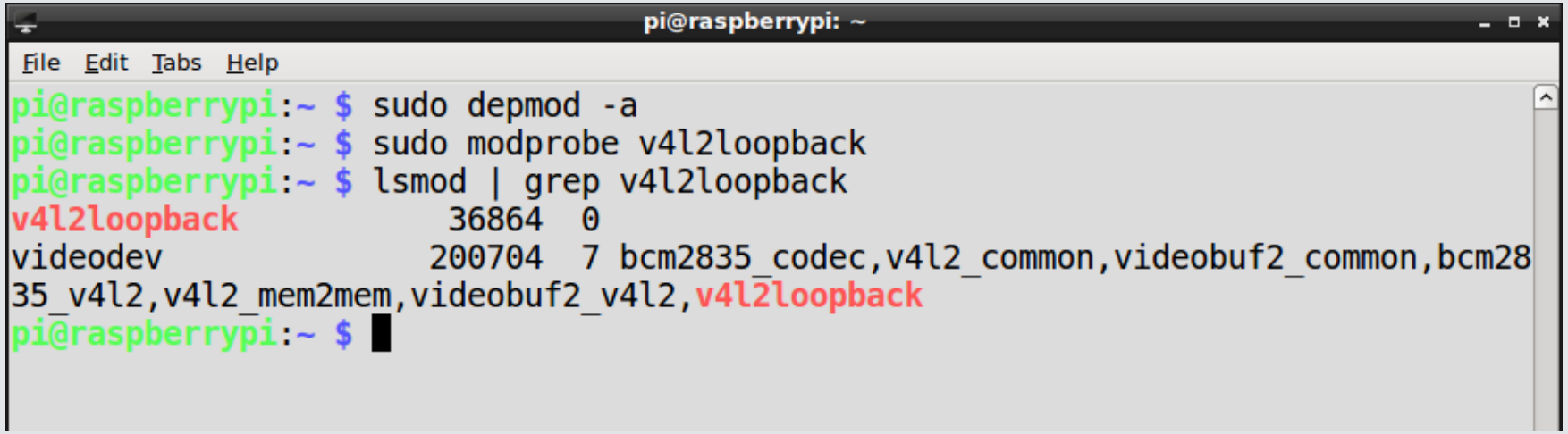
- `$ git clone https://github.com/umlaeute/v4l2loopback`
- `$ cd ~/v4l2loopback`
- `$ sudo make`
- `$ sudo make install`

# 3.安裝V4L2 Kernel Module

- `$ sudo depmod -a`

這三行每次開機都需要重新啟用

- `$ sudo modprobe v4l2loopback`
- `$ lsmod | grep v4l2loopback`




https://github.com/raspberrypi-tw/meetup19

# 4.將SPI 讀取到的影像導到V4L2 裝置

- `$ cd ~/LeptonModule/software/v4l2lepton`

  ==> 可能需要先修改 `v4l2lepton.cpp`

- `$ make`
- `$ sudo ./v4l2lepton -d /dev/spidev0.0 /dev/video1`

  `Waiting for sink`

  `done reading, resets:`

要看 v4l2lepton 裡的 /dev/video 決定 X
已經有相機了，故選1

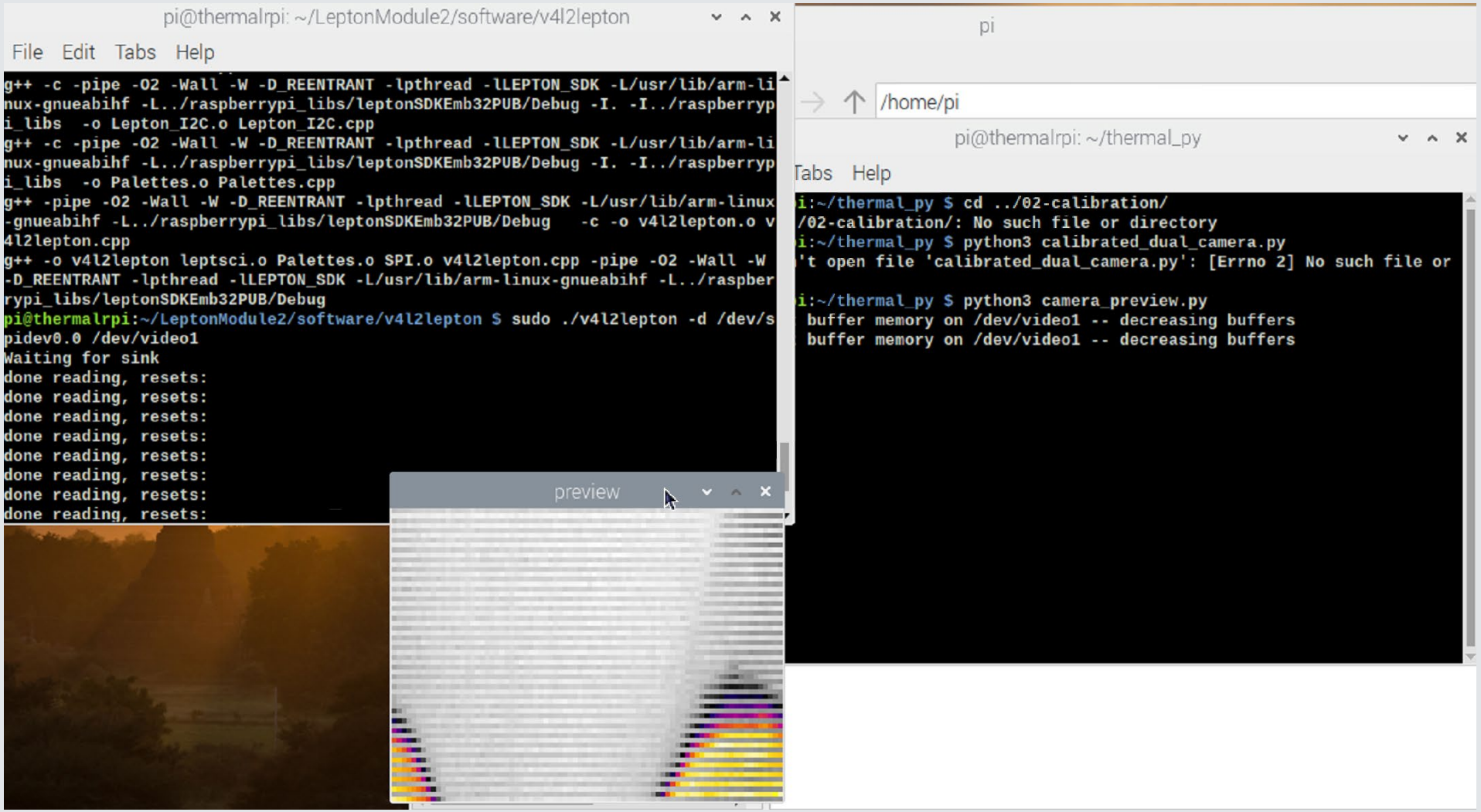
# 再次執行camera_preview.py

```
import cv2
import imutils

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    frame = imutils.resize(frame, 320)
    cv2.imshow("preview", frame)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

cap.release()
cv2.destroyAllWindows()
```

# DEMO
# camera_preview.py

```
$ cd ~/FLIR/thermal-pi/01-
flir
$ python3 camera_preview.py
```

# 執行結果 ( 開兩個視窗)


pi@thermalrpi: ~/LeptonModule2/software/v4l2lepton
File Edit Tabs Help
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -lpthread -lLEPTON_SDK -L/usr/lib/arm-linux-gnueabihf -L../raspberrypi_libs/leptonSDKEmb32PUB/Debug -I. -I../raspberrypi_libs -o Lepton_I2C.o Lepton_I2C.cpp
g++ -c -pipe -O2 -Wall -W -D_REENTRANT -lpthread -lLEPTON_SDK -L/usr/lib/arm-linux-gnueabihf -L../raspberrypi_libs/leptonSDKEmb32PUB/Debug -I. -I../raspberrypi_libs -o Palettes.o Palettes.cpp
g++ -pipe -O2 -Wall -W -D_REENTRANT -lpthread -lLEPTON_SDK -L/usr/lib/arm-linux-gnueabihf -L../raspberrypi_libs/leptonSDKEmb32PUB/Debug -c -o v4l2lepton.o v4l2lepton.cpp
g++ -o v4l2lepton leptsci.o Palettes.o SPI.o v4l2lepton.cpp -pipe -O2 -Wall -W -D_REENTRANT -lpthread -lLEPTON_SDK -L/usr/lib/arm-linux-gnueabihf -L../raspberrypi_libs/leptonSDKEmb32PUB/Debug
pi@thermalrpi:~/LeptonModule2/software/v4l2lepton $ sudo ./v4l2lepton -d /dev/spidev0.0 /dev/video1
Waiting for sink
done reading, resets:
done reading, resets:
done reading, resets:
done reading, resets:
done reading, resets:
done reading, resets:
done reading, resets:
done reading, resets:
preview
pi
/home/pi
pi@thermalrpi: ~/thermal_py
Tabs Help
i:~/thermal_py $ cd ../02-calibration/
/02-calibration/: No such file or directory
i:~/thermal_py $ python3 calibrated_dual_camera.py
't open file 'calibrated_dual_camera.py': [Errno 2] No such file or
i:~/thermal_py $ python3 camera_preview.py
buffer memory on /dev/video1 -- decreasing buffers
buffer memory on /dev/video1 -- decreasing buffers

$ cd ~

# 對應 lepton3 的版本

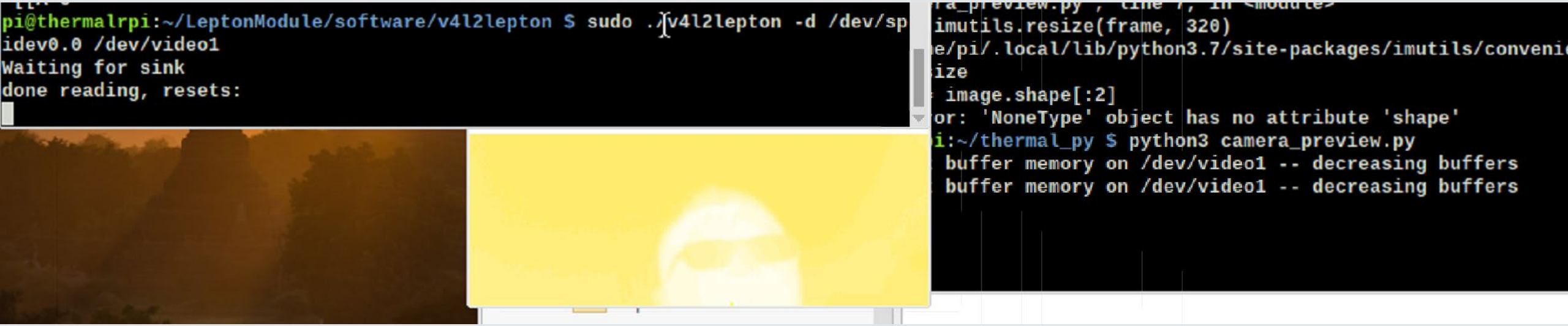
$ git clone https://github.com/acqxi/FLIR.git

---

將裡面的 v4l2lepton.cpp 拖到

/home/pi/LeptonModule/software/v4l2lepton

覆蓋後重新 $ make 與 之前的步驟



# 使用Python串接FLIR Lepton

# pylepton

- Pure python library for capturing images from the Lepton over SPI

https://github.com/groupgets/pylepton

# 安装pylepton

- `$ cd ~`

---

- `$ git clone https://github.com/groupgets/pylepton -b lepton3-dev`
- `$ cd ~/pylepton`
- `$ sudo python3 setup.py install`

# 測試 pylepton_capture 範例程式

- `$ cd ~/pylepton`

---

- `$ ./pylepton_capture output.jpg`
- `$ gpicview output.jpg`

# pylepton_capture

```
import numpy as np
import cv2
from pylepton import Lepton

with Lepton() as l:
    a,_ = l.capture()
cv2.normalize(a, a, 0, 65535, cv2.NORM_MINMAX)
np.right_shift(a, 8, a) # fit data into 8 bits
cv2.imwrite("output.jpg", np.uint8(a))
```

raw sensor data

將 12-bit 的輸出正規化到 OpenCV uint16

# DEMO
# pylepton_capture

```
$ cd ~/pylepton
$ ./pylepton_capture output.jpg
```

# 測試pylepton_overlay 範例程式

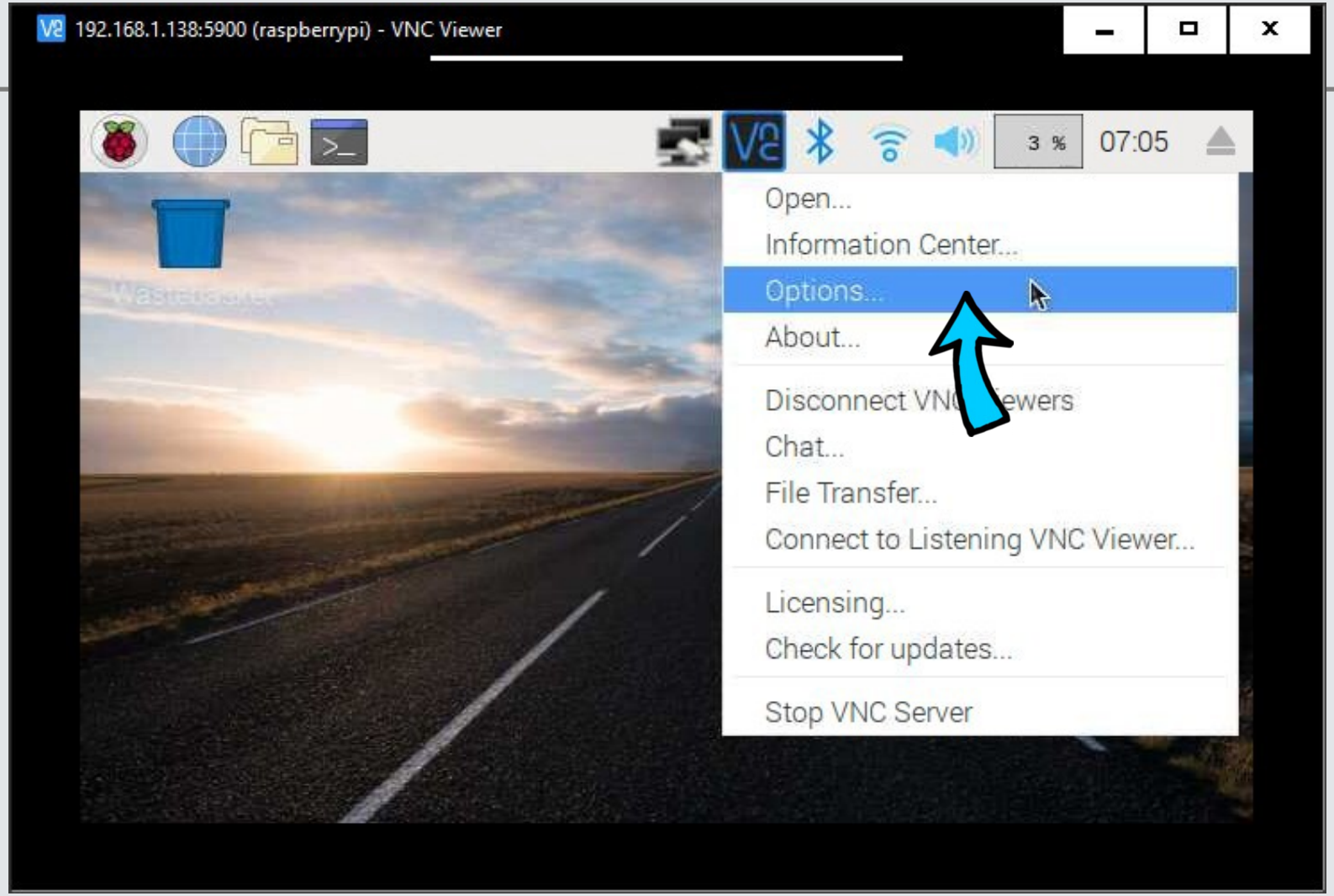
- `$ cd ~/pylepton`

---

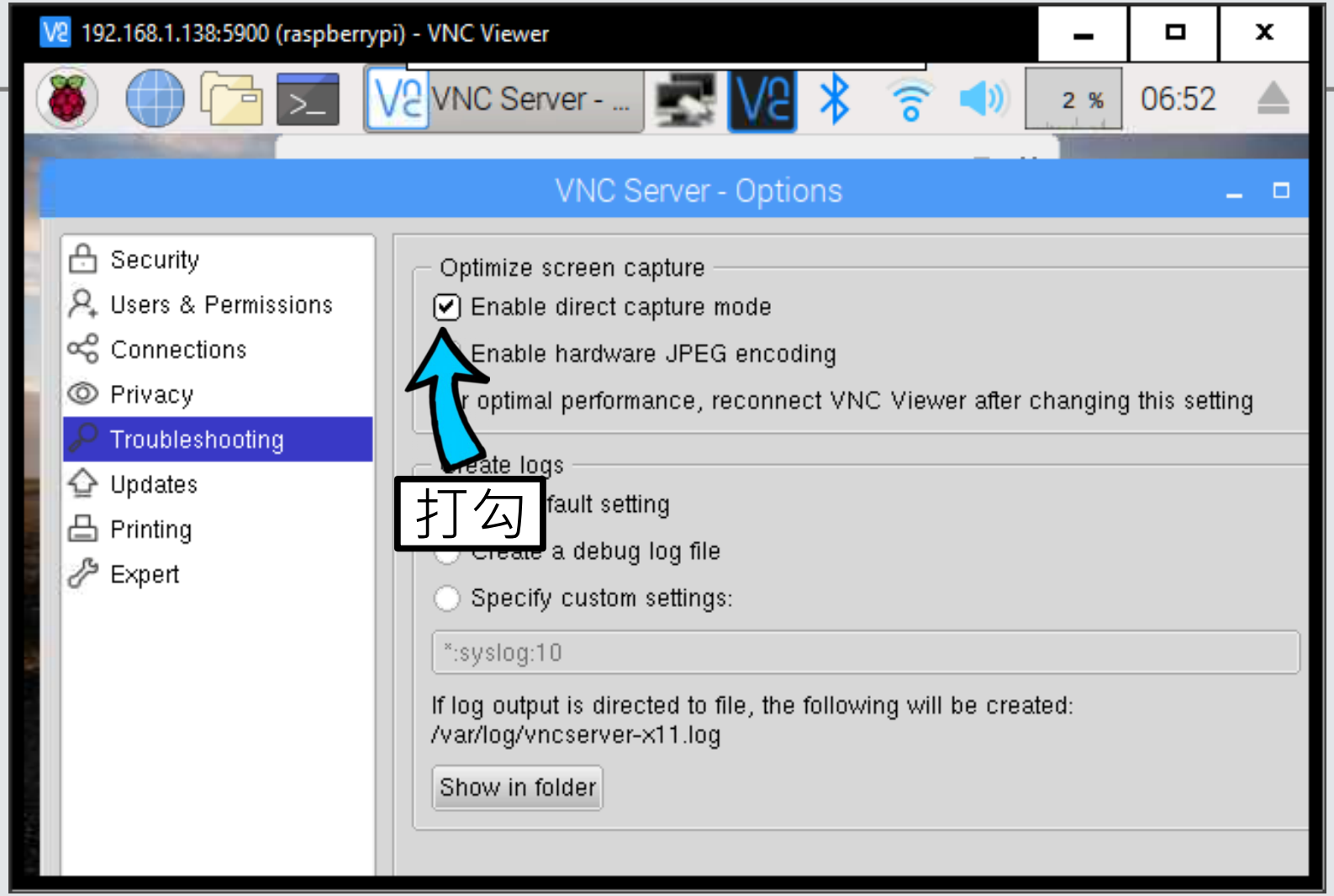
- `$ ./pylepton_overlay`

  ( 需要搭配螢幕或是 VNC Direct Capture Mode 觀察結果 )

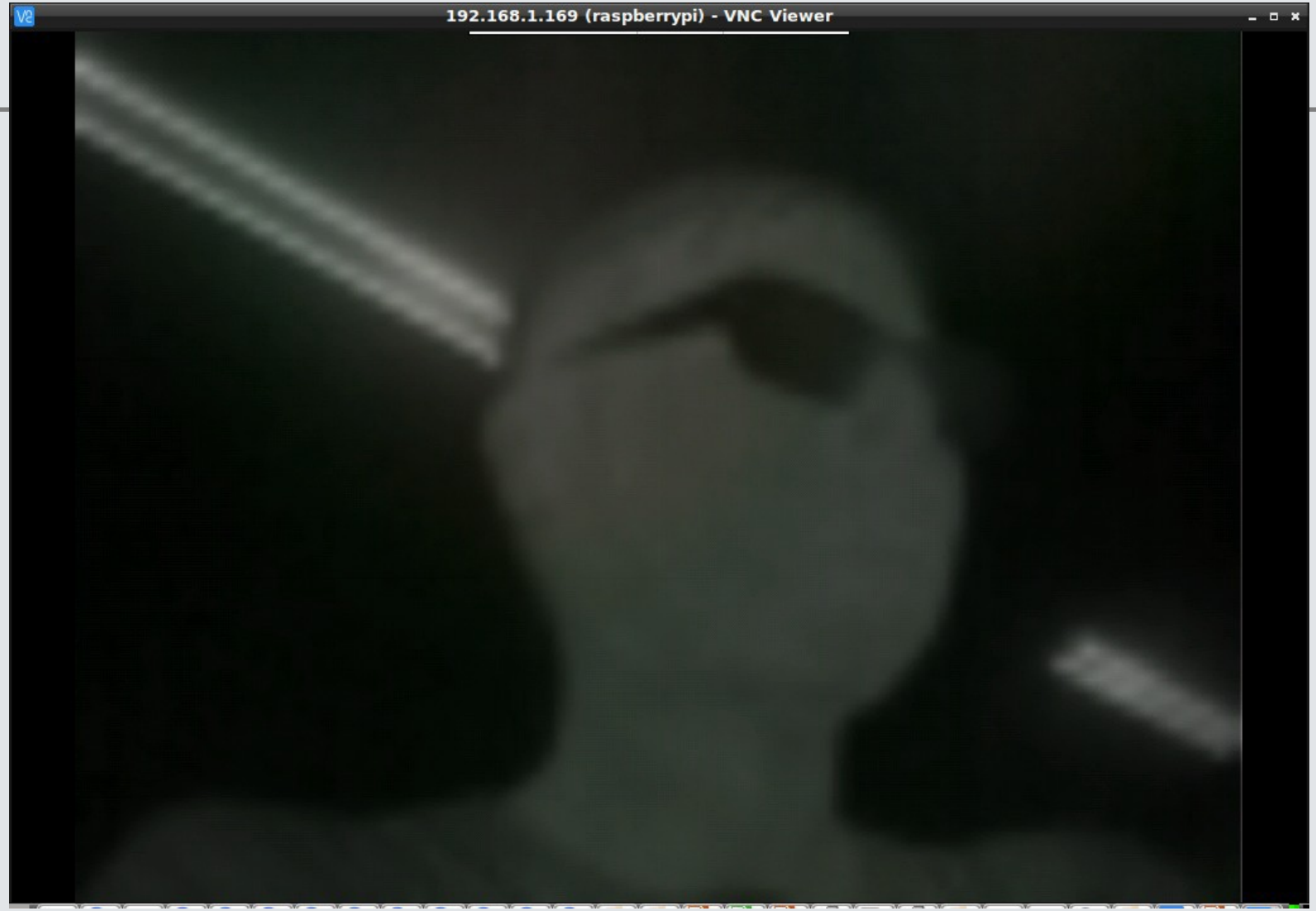
# 修改VNC 設定


192.168.1.138:5900 (raspberrypi) - VNC Viewer
3 %
07:05
Open...
Information Center...
Options...
About...
Disconnect VNC Viewers
Chat...
File Transfer...
Connect to Listening VNC Viewer...
Licensing...
Check for updates...
Stop VNC Server

# VNC Direct Capture Mode



# 執行 ./pylepton_overlay 就可以看到畫面



# pylepton_overlay

```
lepton_buf = np.zeros((60, 80, 1), dtype=np.uint16)

with Lepton(device) as l:
    last_nr = 0
    while True:
        _,nr = l.capture(lepton_buf)
        if nr == last_nr:
            # no need to redo this frame
            continue
        last_nr = nr
        cv2.normalize(lepton_buf, lepton_buf, 0, 65535, cv2.NORM_MINMAX)
        np.right_shift(lepton_buf, 8, lepton_buf)
```

# 將Lepton 讀到的資料轉換後給OpenCV

```
from pylepton import Lepton

with Lepton() as l:
    while True:
        a, _ = l.capture()
        cv2.normalize(a, a, 0, 65535, cv2.NORM_MINMAX)
        np.right_shift(a, 8, a)

        cv2.imshow("preview", np.uint8(a))
        if cv2.waitKey(1) & 0xFF == ord("q"):
            break

        time.sleep(0.01)

cv2.destroyAllWindows()
```

# DEMO
# pylepton_preview.py

```
$ cd ~/FLIR/thermal-pi/01-flir
$ python3 pylepton_preview.py
```

# 輻射溫度換算

https://www.flir.com/discover/security/radiometric/the-benefits-and-challenges-of-radiometric-thermal-technology/

# Radiometry

- Radiometry 是量測電磁光譜的科學
- 熱像機透過紅外光訊號的強度來測量表面溫度
- FLIR Lepton Radiometry 系列可更精確量測紅外線輻射強度，並轉換成校正過的溫度

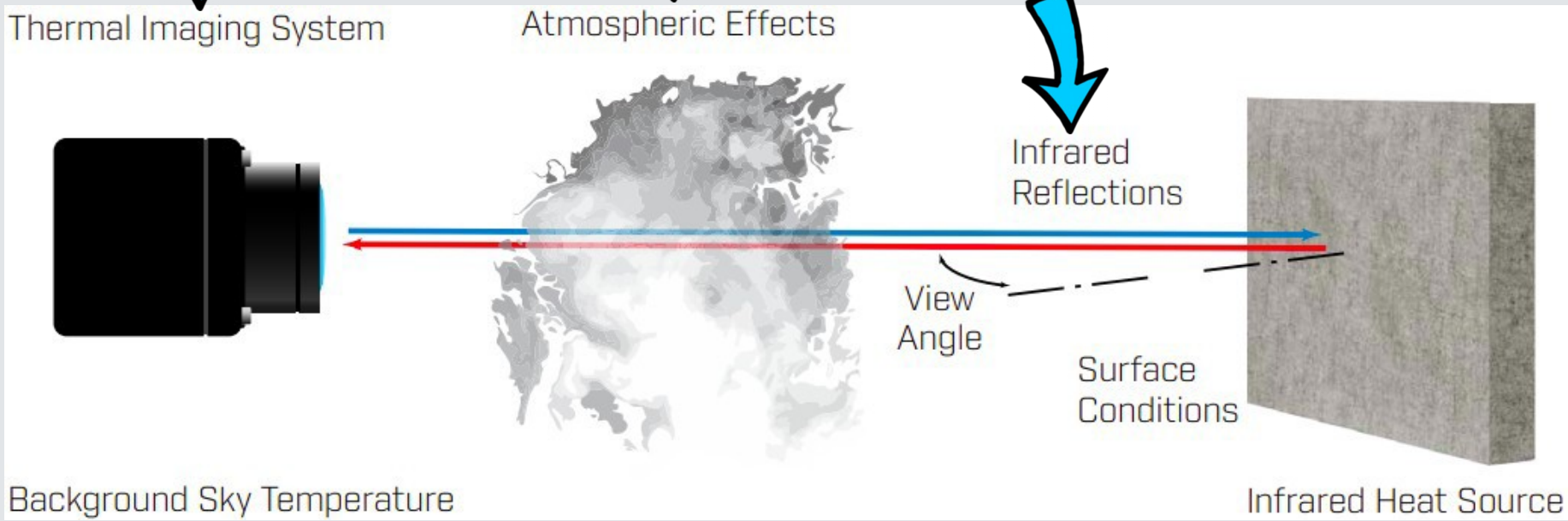
https://lepton.flir.com/radiometry/

# 影響Radiometry 溫度量測的原因

解析度 (Resolution)

大氣影響 (Atmosphere)

反射率 (Reflectivity)

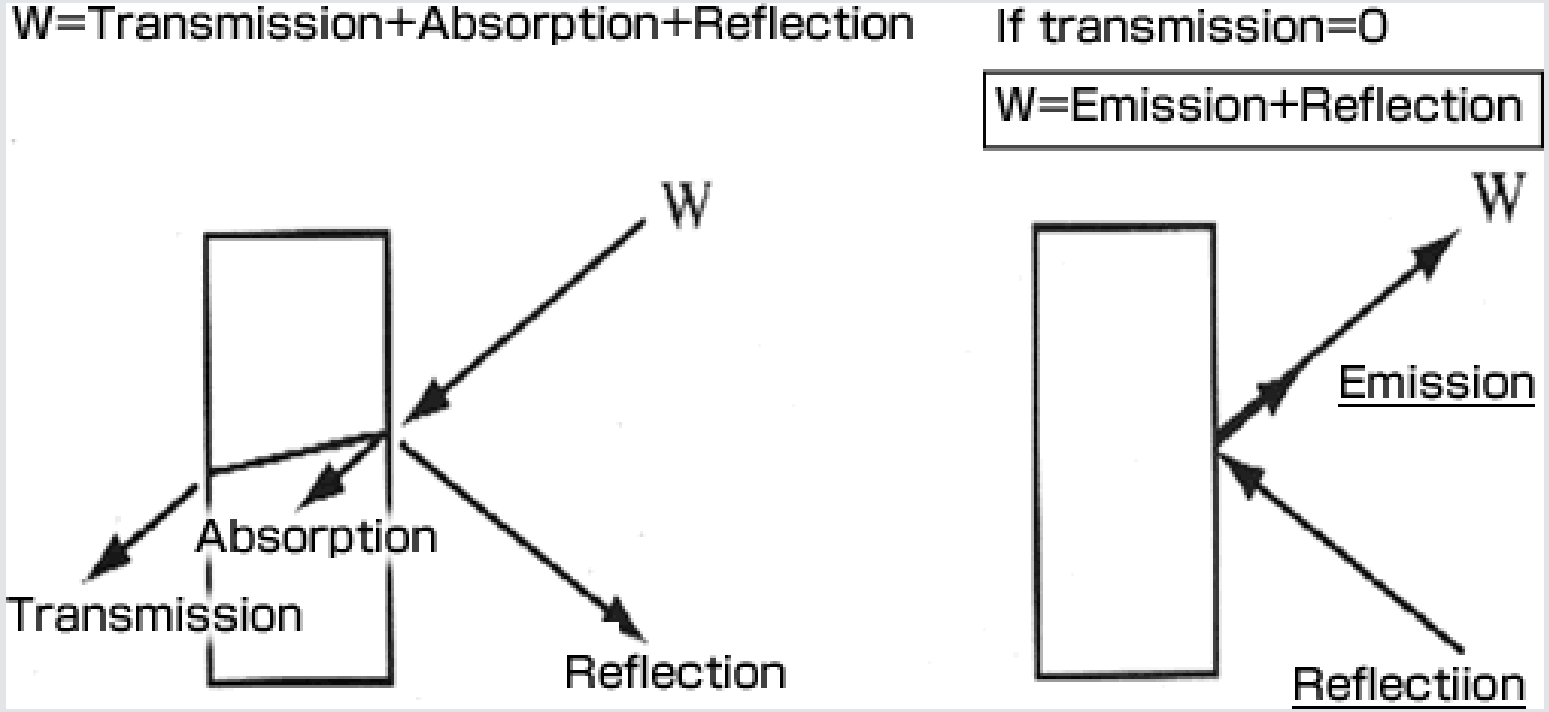



http://bit.ly/3hrIAZ5

發射率 (Emissivity)

# Kirchhoff's Law of Thermal Radiation

- 在熱平衡下，物體輻射的功率等於吸收的功率
- 輻射包含發射，反射和透射

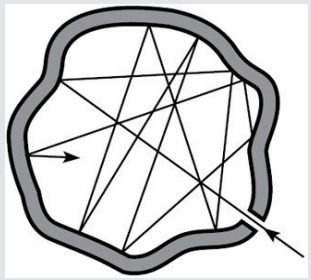



http://www.infrared.avio.co.jp/en/products/ir-thermo/what-thermo.html

# 發射率 (Emissivity)

- 發射率量化物體表面在特定波段和特定溫度下輻射能量的效率
- 完美輻射 (100%)
- 無輻射 (0%)
- 可用黑體校正輻射量值


黑體校正儀器



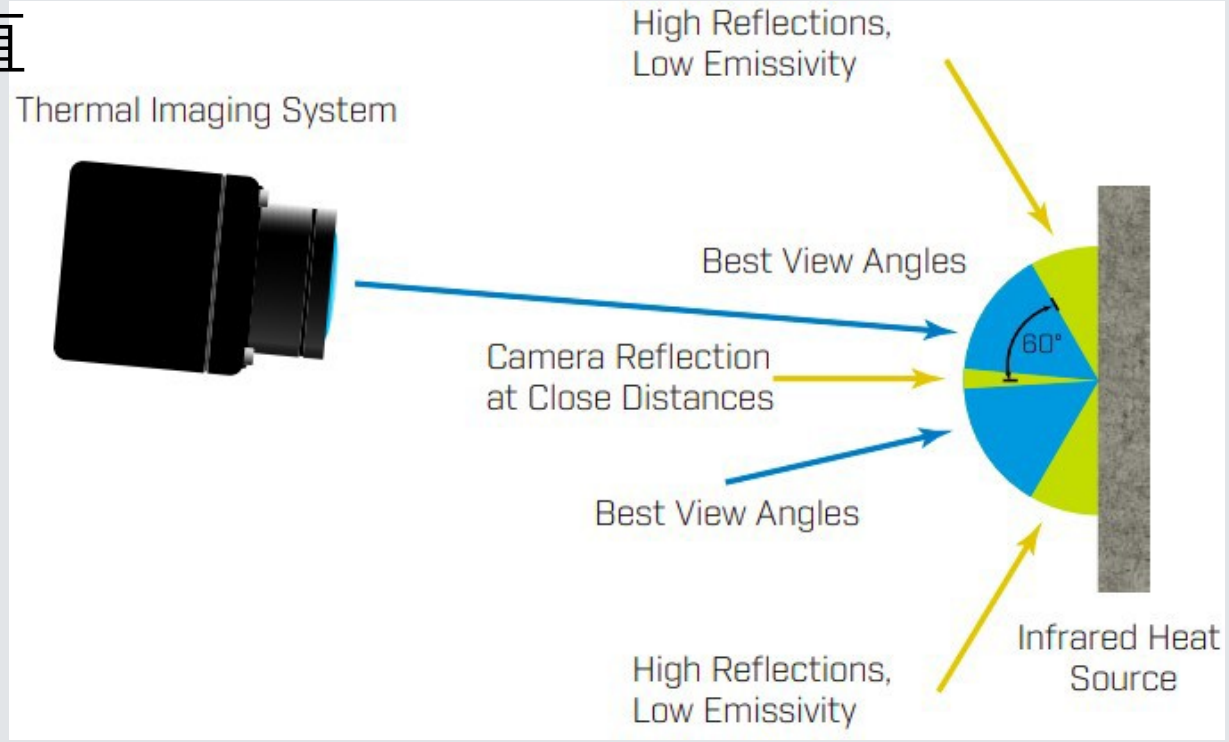
黑體校正原理

http://bit.ly/2Blbx9Z

| Material Description | Emissivity [n.d.] [1, 2] |
|---|---|
| Asphalt | 0.90 to 0.98 |
| Concrete | 0.92 |
| Soil, dry | 0.90 |
| Soil, wet | 0.95 |
| Wood | 0.90 |
| Water | 0.92 to 0.96 |
| Ice | 0.96 to 0.98 |
| Snow | 0.83 |
| Brick | 0.93 to 0.96 |
| Lacquer, paint | 0.80 to 0.95 |
| Lacquer, flat black | 0.97 |
| Textiles | 0.90 |
| Skin, human | 0.98 |
| Aluminum, polished | 0.04 to 0.06 |
| Aluminum, anodised | 0.55 |
| Steel, rusty | 0.69 |
| Steel, stainless | 0.16 to 0.45 |

# 反射率 (Reflectivity)

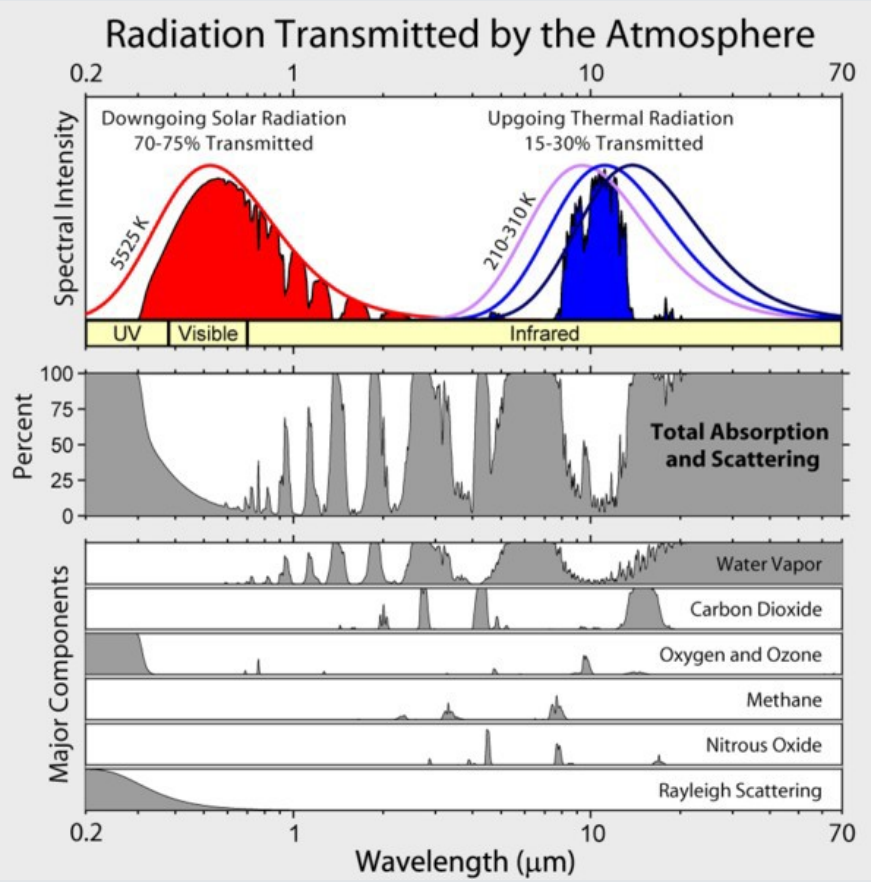
- E+R=1,(R= 反射率 ,E= 發射率 )
- 校正溫度表示總能量有多少百分比是從表面輻射
- 反射率和材質，角度有關
- 觀測角不為垂直
- 最佳角度 <60°




http://bit.ly/39mavXF

# 大氣影響 (Atmosphere)

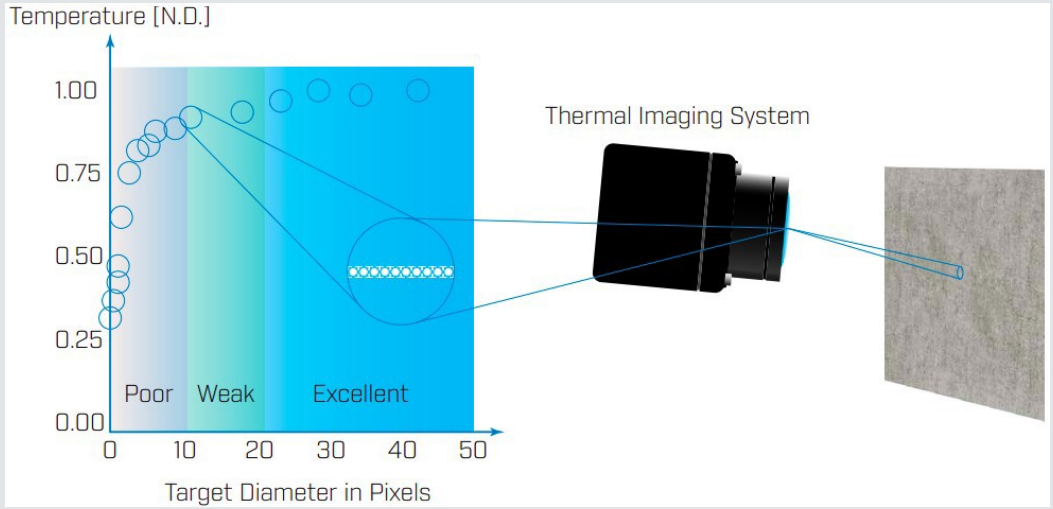
- 空氣密度，空氣溫度，相對濕度影響紅外線輻射



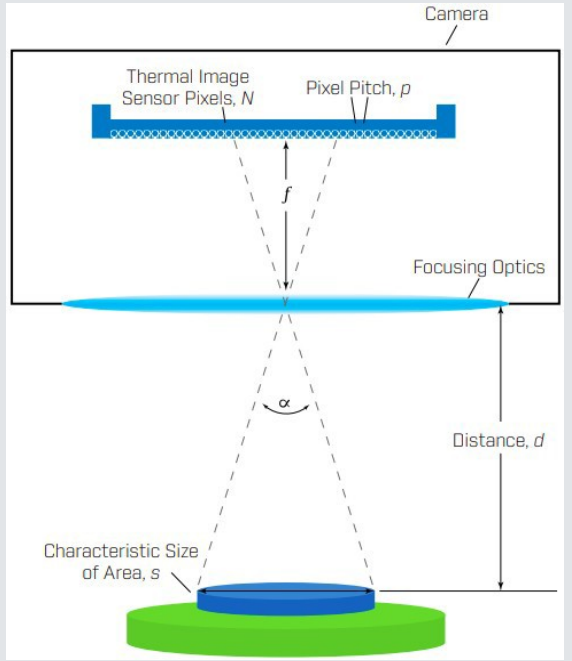

https://en.wikipedia.org/wiki/Absorption_band

# 解析度(Resolution)

- 輻射溫度精確度和解析度有直接關係
- 測量物直徑為 10 pixel 才有測量意義
- 物體表面積的像素數量和像素間距，焦距，距離等有關



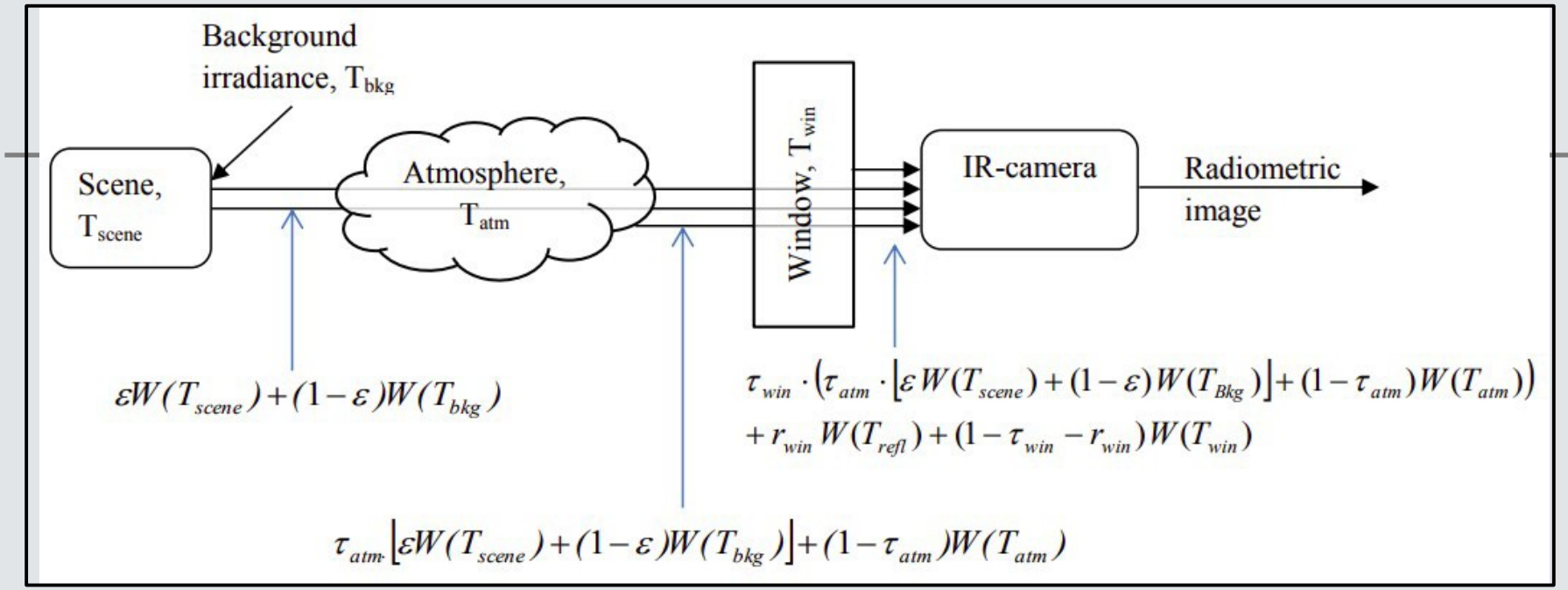

有效測量




物體表面積和像素換算

http://bit.ly/3hrIAZ5

# 溫度計算模型與公式




溫度計算模型

$$S = \tau_{win} \cdot \left(\tau_{atm} \cdot \left[\varepsilon W(T_{scene}) + (1-\varepsilon) W(T_{Bkg})\right] + (1-\tau_{atm}) W(T_{atm})\right) + r_{win} W(T_{refl}) + (1-\tau_{win}-r_{win}) W(T_{win})$$

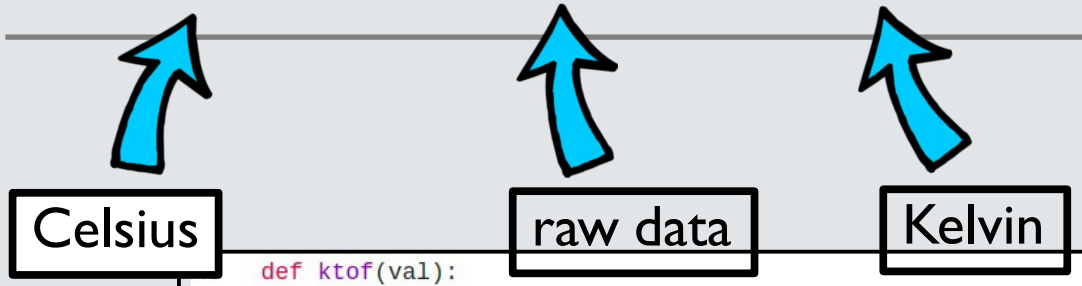
溫度計算公式

| Notation | Description |
|---|---|
| $S$ | Value of the 14-bit digital video in counts |
| $\varepsilon$ | Emissivity of the scene. |
| $\tau_{win}$ | Transmission coefficient of the window |
| $T_{win}$ | Window temperature |
| $r_{win}$ | Window reflection |
| $T_{refl}$ | Temperature reflected in the window |
| $\tau_{atm}$ | Transmission coefficient of the atmosphere between the scene and the camera |
| $T_{atm}$ | Atmospheric temperature |
| $T_{bkg}$ | Background temperature (reflected by the scene) |
| $T_{scene}$ | Scene temperature |
| $W(T)$ | Radiated flux (in units of counts) as function of the |

http://bit.ly/39hAr6U

# 還好 FLIR Lepton 都幫你算好了！

- `t(℃) = (raw - 27315) / 100.0`




```
def ktof(val):
41    return (1.8 * ktoc(val) + 32.0)
42
43  def ktoc(val):
44    return (val - 27315) / 100.0
45
46  def raw_to_8bit(data):
47    cv2.normalize(data, data, 0, 65535, cv2.NORM_MINMAX)
48    np.right_shift(data, 8, data)
49    return cv2.cvtColor(np.uint8(data), cv2.COLOR_GRAY2RGB)
50
51  def display_temperature(img, val_k, loc, color):
52    val = ktof(val_k)
53    cv2.putText(img,"{0:.1f} degF".format(val), loc, cv2.FONT_HERSHEY_SIMPLEX, 0.75, color, 2)
54    x, y = loc
55    cv2.line(img, (x - 2, y), (x + 2, y), color, 1)
56    cv2.line(img, (x, y - 2), (x, y + 2), color, 1)
```

https://github.com/groupgets/purethermal1-uvc-capture/blob/master/python/uvc-radiometry.py

# 印出 [3][3] 的溫度

```
from pylepton import Lepton

with Lepton() as l:
    while True:
        lepton_buf, nr = l.capture()
        lepton_temp = np.copy(lepton_buf)
        a = np.copy(lepton_buf)
        cv2.normalize(a, a, 0, 65535, cv2.NORM_MINMAX)
        np.right_shift(a, 8, a)
        ...
        t = lepton_temp[3][3]
        print(t, int((t - 27315)/100))

        cv2.imshow("temperature", _lepton_gray)
        time.sleep(0.01)

cv2.destroyAllWindows()
```

# 轉換ColorMap

```
_lepton = np.asarray(lepton_buf, np.uint8)
_lepton_gray = cv2.cvtColor(_lepton, cv2.COLOR_GRAY2RGB)
_lepton_gray = cv2.resize(_lepton_gray, (w, h))
_lepton_gray = cv2.applyColorMap(_lepton_gray, cv2.COLORMAP_JET)
#_lepton_gray = cv2.applyColorMap(_lepton_gray,
cv2.COLORMAP_RAINBOW)
```

# DEMO
# pylepton_get_temp.py

```
$ cd ~/FLIR/thermal-pi/01-flir
$ python3 pylepton_get_temp.py
```

# 使用新版 pylepton

將程式中的 *from* pylepton *import* Lepton 與 *with* Lepton() *as*

```
from pylepton import Lepton

cap = cv2.VideoCapture(0)

try:
    with Lepton() as l:
        while True:
```

```
from pylepton.Lepton3 import Lepton3

cap = cv2.VideoCapture(0)

try:
    with Lepton3() as l:
        while True:
```

之後使用的每份py檔案都需要改，
才能使用lepton3

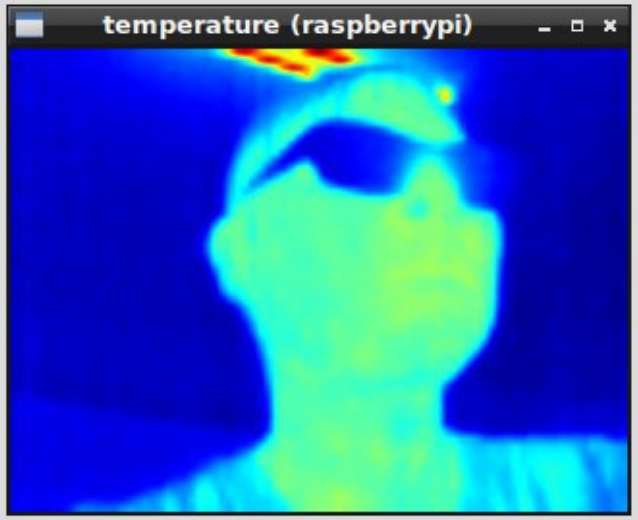
改成 *from* pylepton.Lepton3 *import* Lepton3 與 *with* Lepton3() *as* l:

# 執行結果

pi@raspberrypi: ~/thermal-pi/01-flir

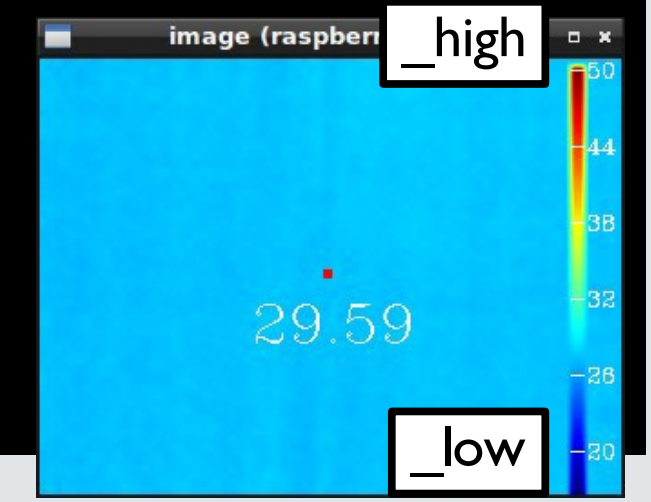
File Edit Tabs Help

```
[30144] 28
[30134] 28
[30156] 28
[30160] 28
[30162] 28
[30178] 28
[30152] 28
[30148] 28
[30160] 28
[30154] 28
[30034] 27
[30079] 27
[30079] 27
[30097] 27
[30131] 28
[30138] 28
[30131] 28
[30140] 28
[30129] 28
[30140] 28
[30127] 28
[30134] 28
[30134] 28
```

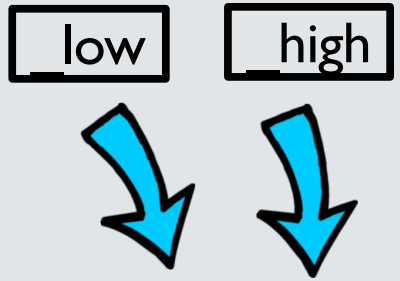


# 畫出溫度條(Temperature Bar)

```
def setColorBar(lepton_buf, _low, _high):
    d = (_high - _low) / 60.0
    i = 0

    for i in range(1, 60):
        _d = d * i
        lepton_buf[i][74] = _high - int(_d)
        lepton_buf[i][73] = _high - int(_d)
```



# DEMO
# pylepton_get_temp.py

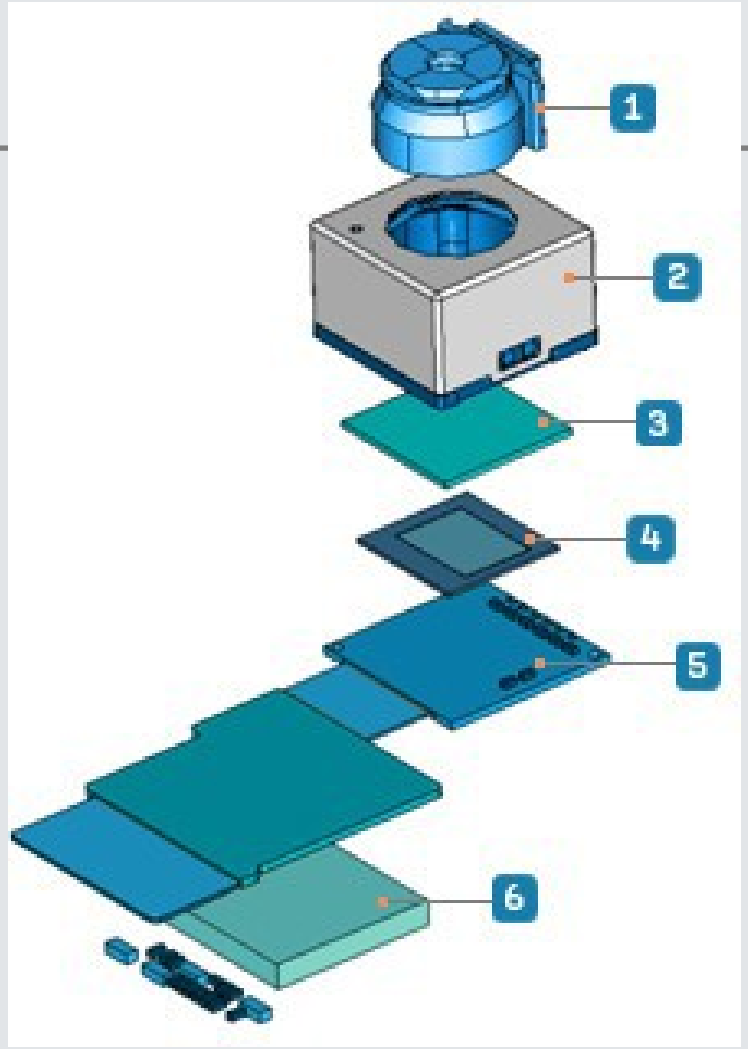



```
$ cd ~/FLIR/thermal-pi/01-flir
$ python3 pylepton_temp_colorbar.py 20 60
```

問題：要如何知道額溫？

# Raspberry Pi Camera簡介

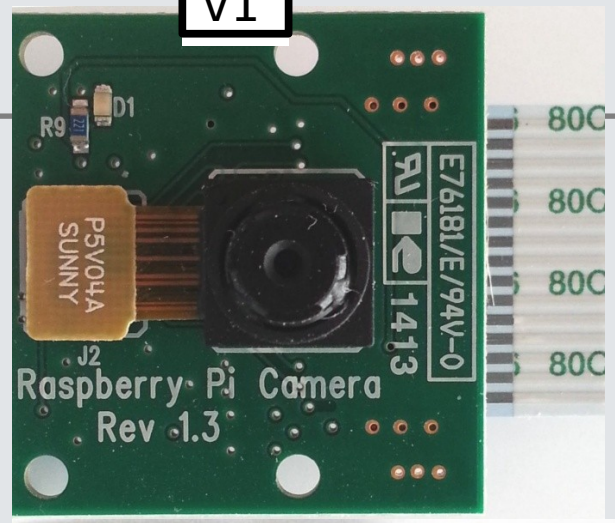
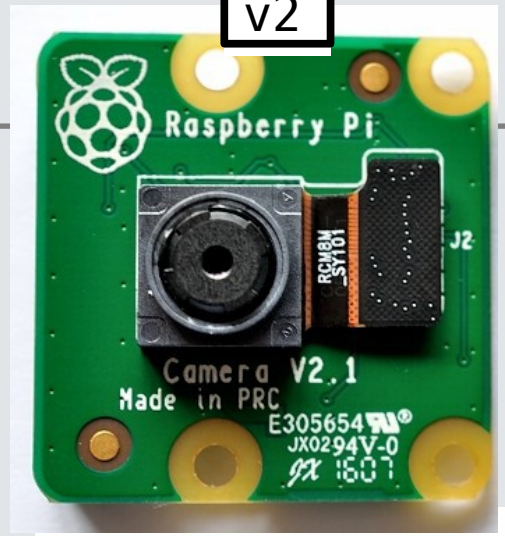
# 從手機相機模組講起




1. Lens( 透鏡 )
2. VCM( 音圈馬達 )
3. IR-Cut( 紅外光濾片 )
4. Sensor( 感光元件 )
5. PCB( 印刷電路板 )
6. ISP( 影像訊號處理器 )

http://www.phonearena.com/news/13MP-camera-tipped-for-Samsung-Galaxy-S-IV_id35168

# Type of Raspberry Pi Camera

| | v1 | v2 |
|---|---|---|
| Raspberry Pi Camera Module |  |  |
| NoIR Camera Module |  |  |



http://elinux.org/Rpi_Camera_Module

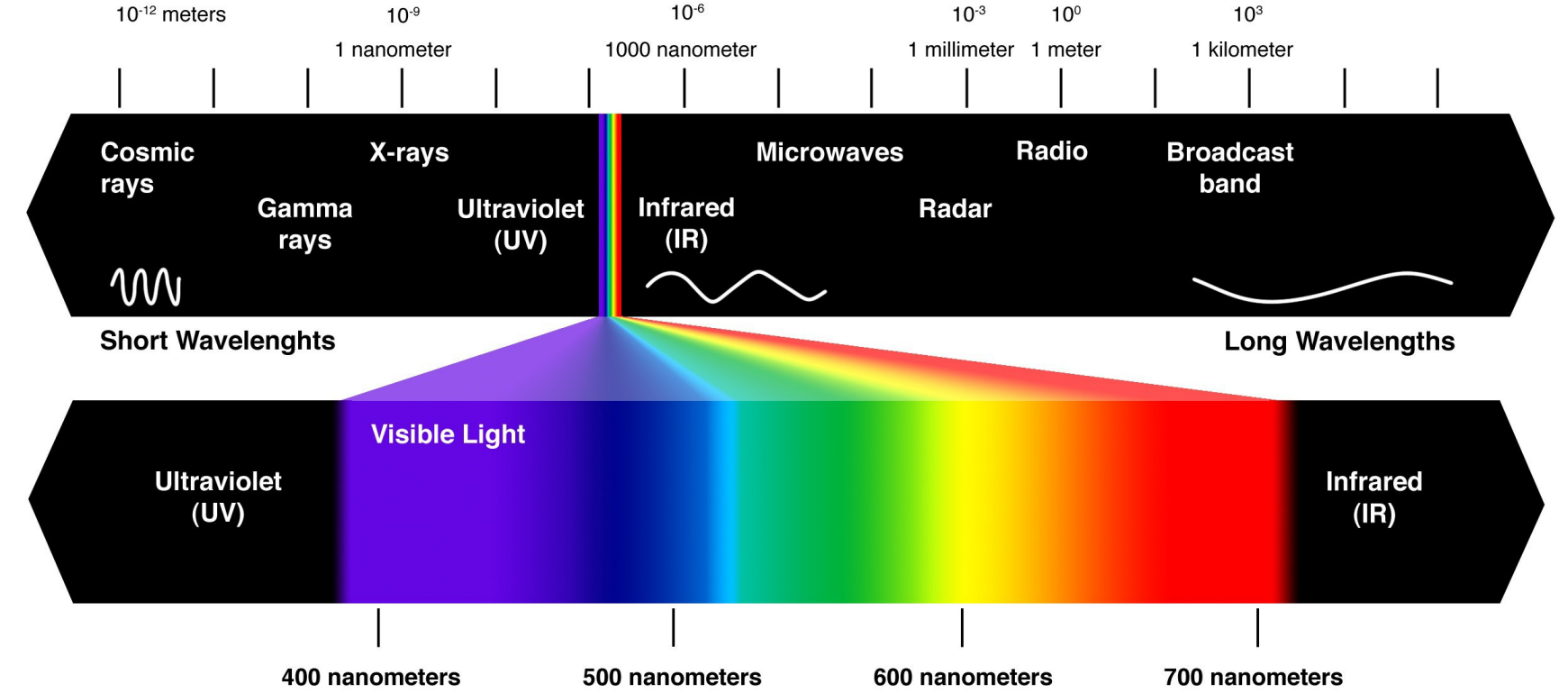
# Raspberry Pi Camera 技術規格(v2)

- Sensor:Sony IMX219 (8MP)
- 靜態拍照最高解析度:3280 x 2464 pixel
- Pixel Size:1.4 x 1.4 µm
- Lens:f=3.6 mm, f/2.9
- Fixed Focus:1m to infinity
- 動態攝影最高解析度:1080p@30 FPS with H.264/AVC

https://www.raspberrypi.org/documentation/hardware/camera/

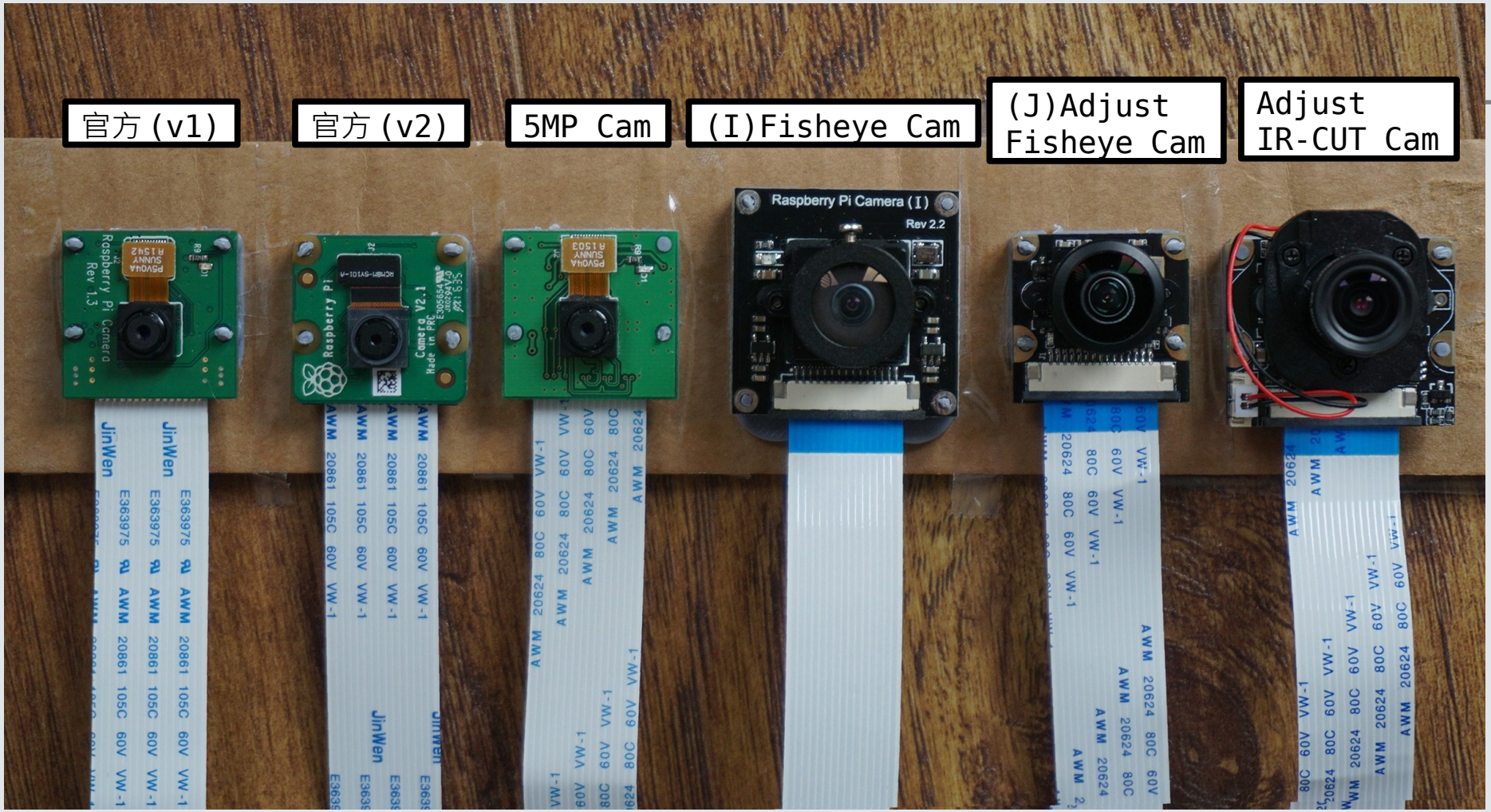
# 基礎光學原理

- 問：樹葉為什麼看起來是綠色的？
- 答：因為樹葉吸收了大部分可見光,只反射綠色光




https://www.raspberrypi.org/learning/infrared-bird-box/worksheet/

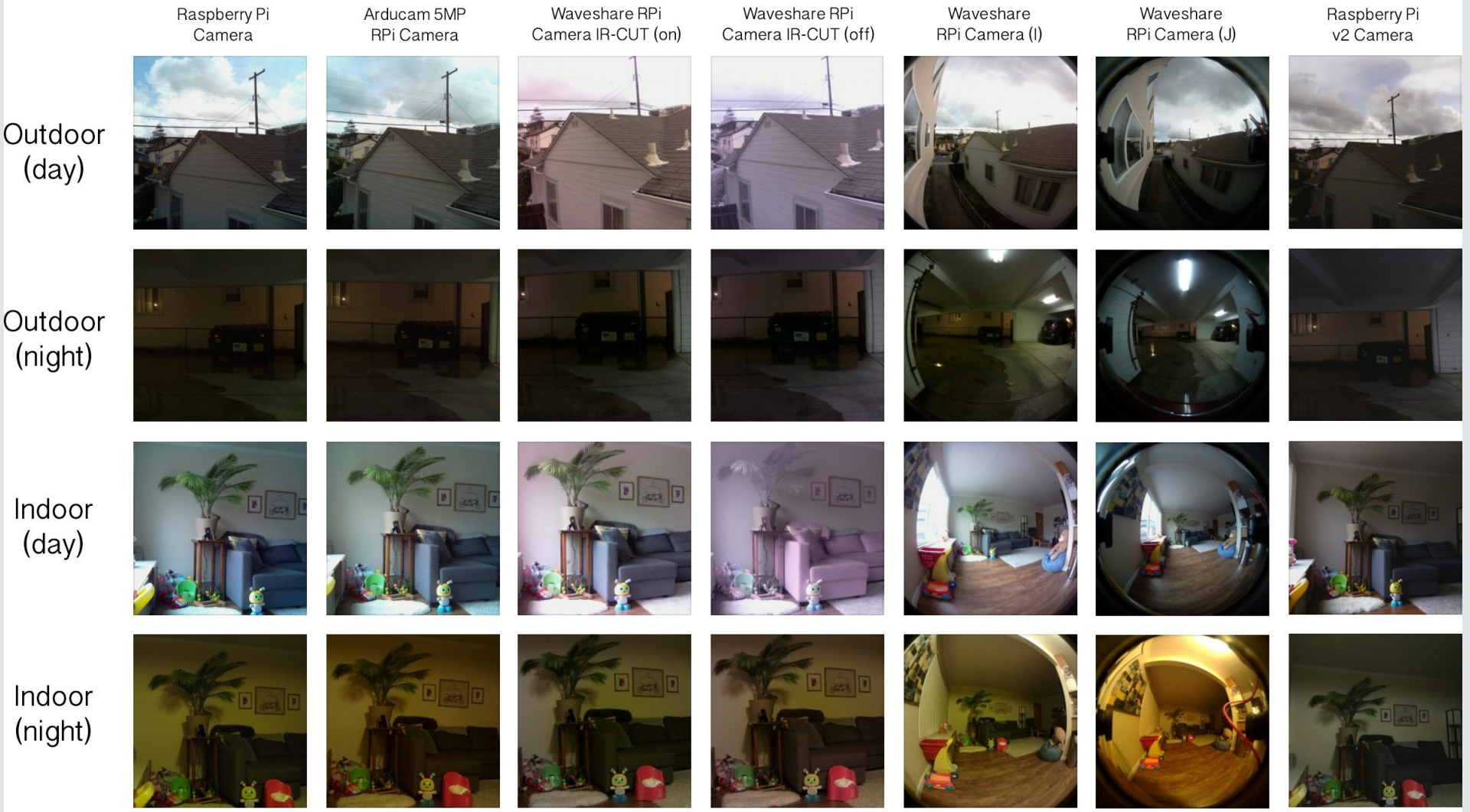
# 更多Camera




http://www.semifluid.com/2017/01/23/raspberry-pi-camera-comparison/

# Raspberry Pi Camera Comparison

semifluid.com



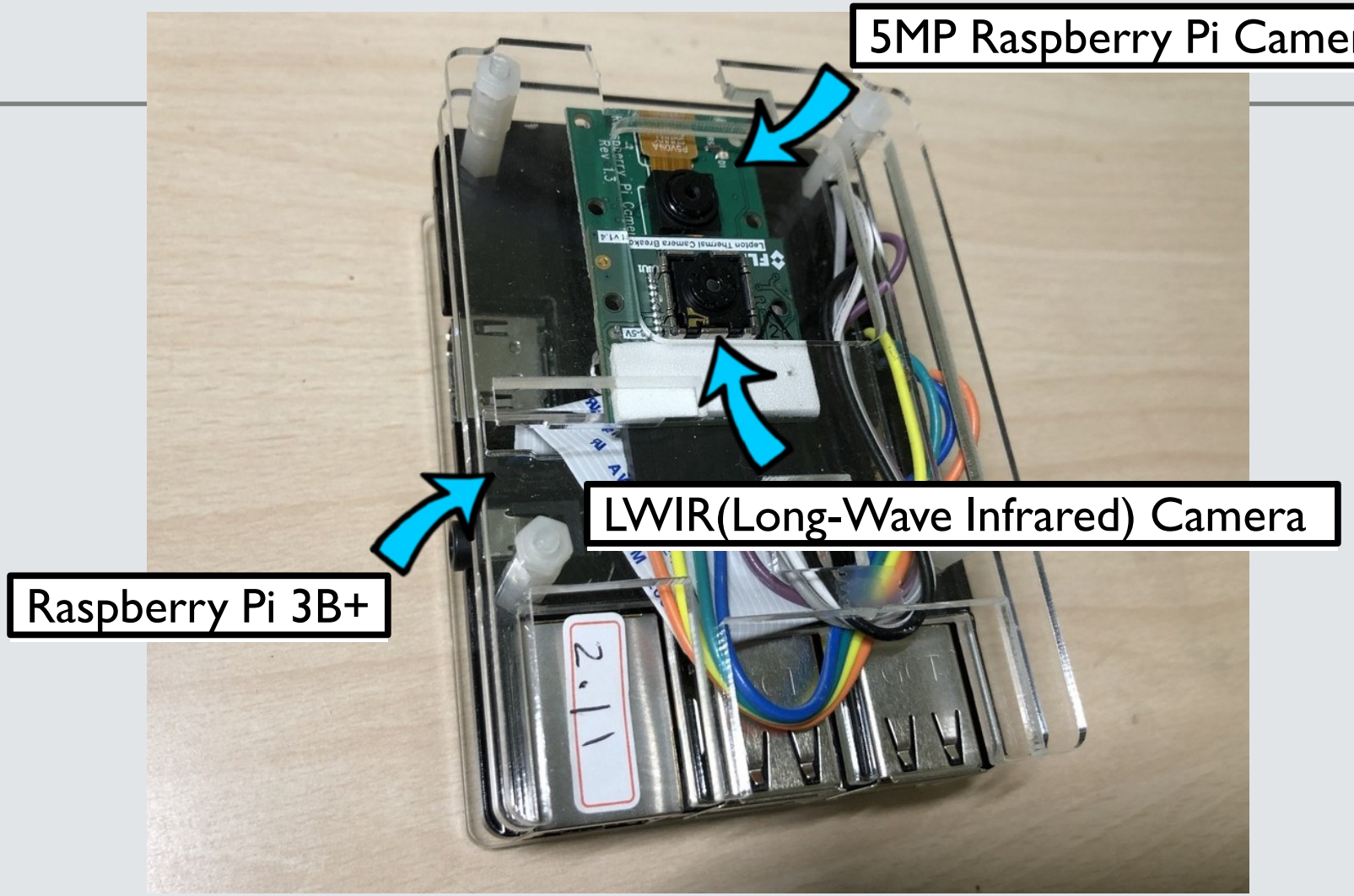

http://www.semifluid.com/2017/01/23/raspberry-pi-camera-comparison/

# Camera安装

# 再次執行camera_preview.py

```
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    frame = imutils.resize(frame, 320)
    cv2.imshow("preview", frame)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

cap.release()
cv2.destroyAllWindows()
```

# 雙相機雛型系統


5MP Raspberry Pi Camera
LWIR(Long-Wave Infrared) Camera
Raspberry Pi 3B+

# 雙相機顯示

```
cap = cv2.VideoCapture(0)

with Lepton() as l:
    while True:
        a, _ = l.capture()
        cv2.normalize(a, a, 0, 65535, cv2.NORM_MINMAX)
        np.right_shift(a, 8, a)
        _a = np.asarray(a, np.uint8)
        _a_rgb = cv2.cvtColor(_a, cv2.COLOR_GRAY2RGB)
        img1 = cv2.resize(_a_rgb, (160, 120), interpolation = cv2.INTER_CUBIC)

        _, img2 = cap.read()
        img2 = imutils.resize(img2, 160)

        horizontal = np.hstack((img1, img2))
        cv2.imshow("dual_camera", horizontal)
```

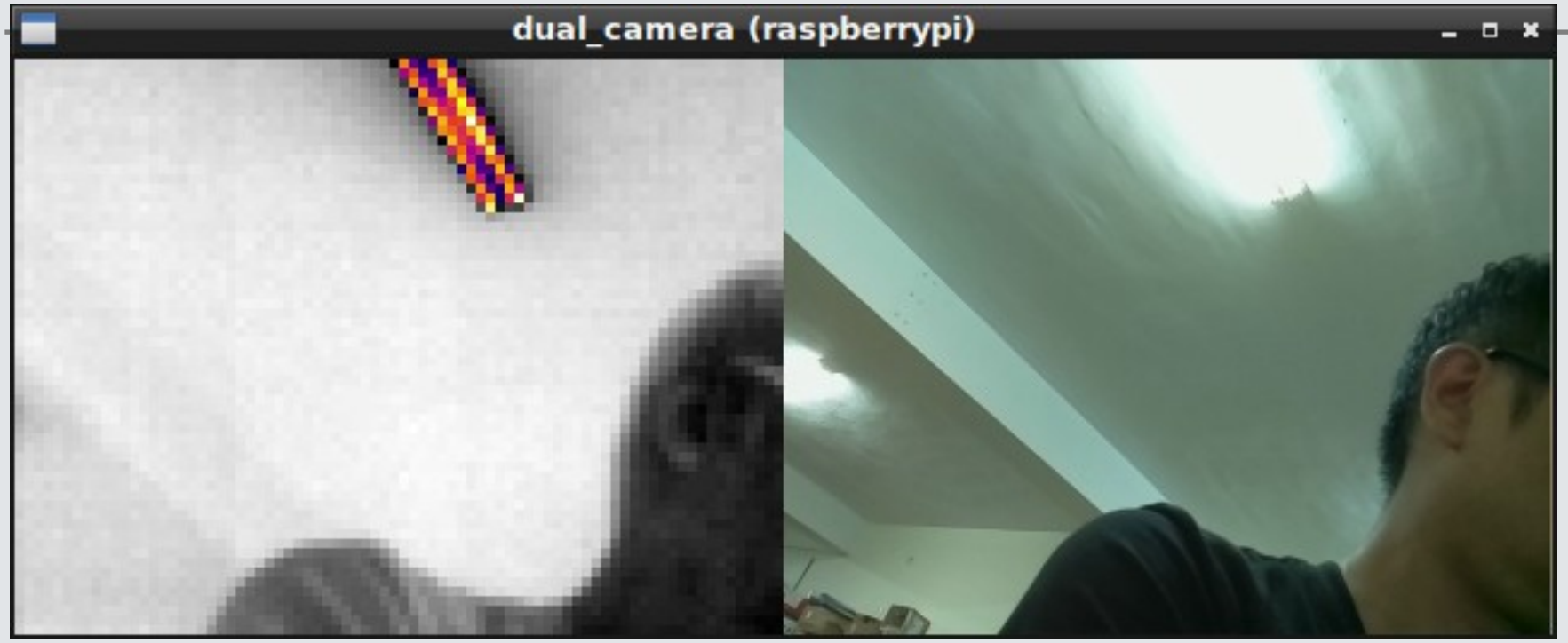
熱相機

光相機

# DEMO
# dual_camera.py

```
$ cd ~/FLIR/thermal-pi/02-
calibration
$ python3 dual_camera.py
```

# 將雙相機並排顯示



# 雙相機alpha blending 混合

```
cv2.namedWindow("blend_camera", cv2.WINDOW_NORMAL)
cv2.createTrackbar("alpha", "blend_camera", 0, 10, nothing)
cap = cv2.VideoCapture(0)

with Lepton() as l:
    while True:
        a, _ = l.capture()
        ...
        img1 = cv2.resize(_a_rgb, (160, 120), interpolation = cv2.INTER_CUBIC)
        _, img2 = cap.read()
        img2 = imutils.resize(img2, 160)

        cv2.resizeWindow("blend_camera", 160, 120)
        visible_alpha = float(alpha)/10
        thermal_alpha = float(10-alpha)/10
        dst = cv2.addWeighted(img1, visible_alpha, img2, thermal_alpha, 0)
        cv2.imshow("blend_camera", dst)
```

熱相機

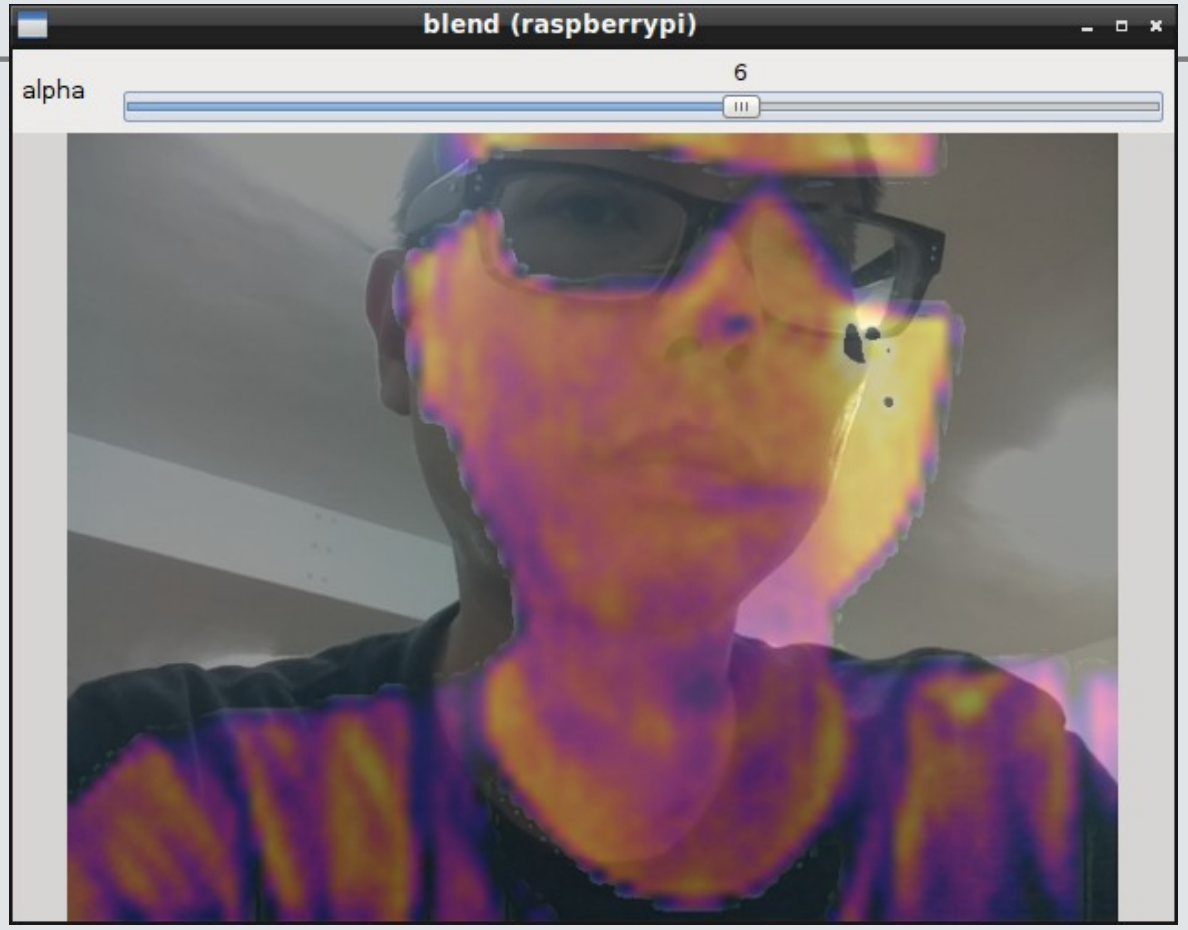
光相機

alpha blending

# DEMO
# blend_camera.py

```
$ cd ~/FLIR/thermal-pi/02-
calibration
$ python3 blend_camera.py
```

# 調整雙相機的alpha 值



# 雙相機校正

# 相機矩陣模型


Lens configuration (internal parameter)

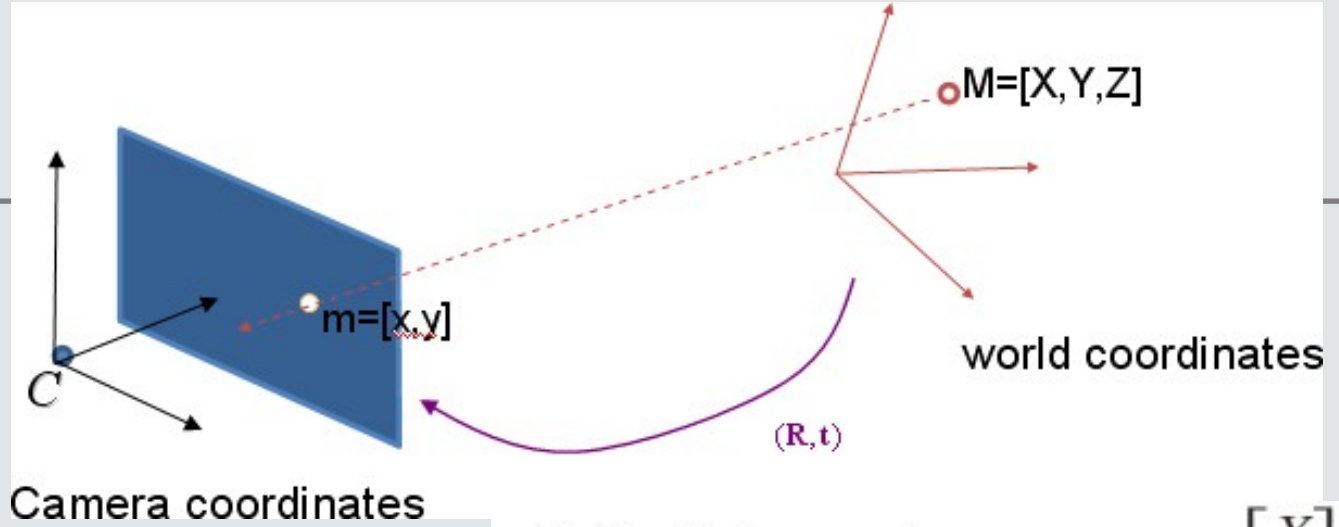
$$\begin{bmatrix} x \\ 1 \end{bmatrix} = L \left( K \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X \\ 1 \end{bmatrix} \right)$$

Spatial relationship between sensor and pinhole (internal parameter)

Camera body configuration (extrinsic parameter)

https://www.coursera.org/lecture/robotics-perception/pinhole-camera-model-wKcXj

# 座標和相機矩陣表示式




$$\underbrace{s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\tilde{m}} = \underbrace{\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix}}_{[R\,t]} \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{\tilde{M}}$$

2D 影像座標

3D 世界座標

http://wycwang.blogspot.com/2012/10/camera-calibration-part-2-calibration.html

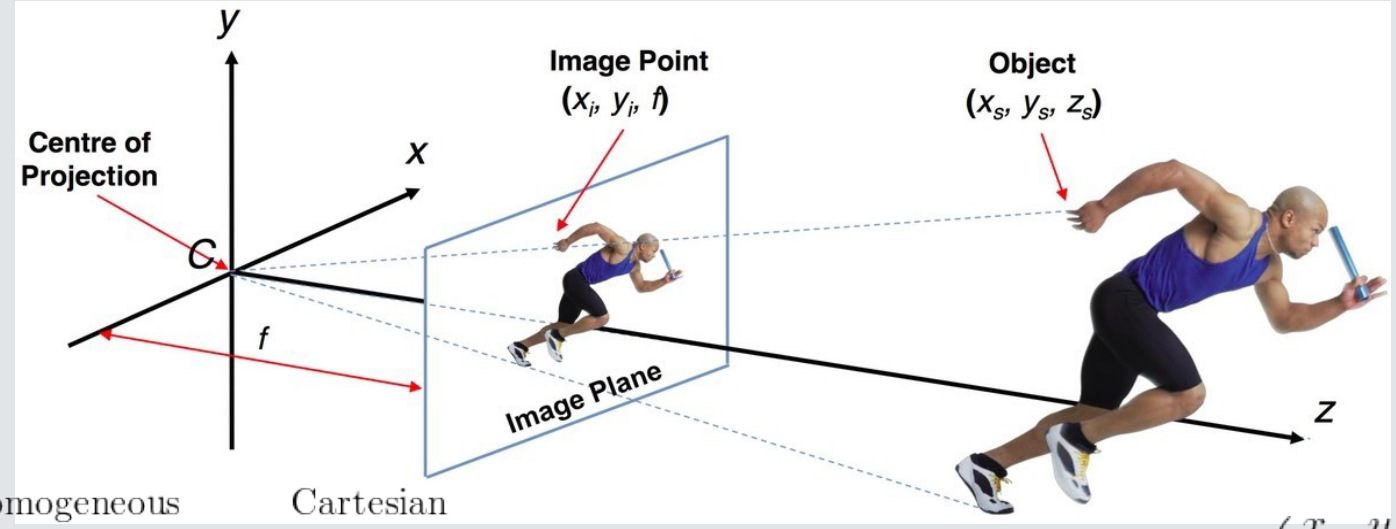
# 3D 轉2D 的問題

- 少了一個距離維度，無法回推真實的世界座標
- 因為歐幾里得或是笛卡爾座標難以描述透視空間
- 例如：在歐幾里得空間中的兩條平行線不能相交

https://attackllama.com/2016/12/homogeneous-coordinates/

# 使用齊次座標表示透視投影結果

- 齊次座標(Homogeneous Coordinates)是用於投影幾何裡的坐標系統。使用N+1維來代表N維座標




Homogeneous Cartesian

$$(1,2,3) \Rightarrow \left(\frac{1}{3},\frac{2}{3}\right)$$

$$(2,4,6) \Rightarrow \left(\frac{2}{6},\frac{4}{6}\right) = \left(\frac{1}{3},\frac{2}{3}\right)$$

$$(4,8,12) \Rightarrow \left(\frac{4}{12},\frac{8}{12}\right) = \left(\frac{1}{3},\frac{2}{3}\right)$$

$$(x,y,w) \Leftrightarrow \left(\frac{x}{w},\frac{y}{w}\right)$$

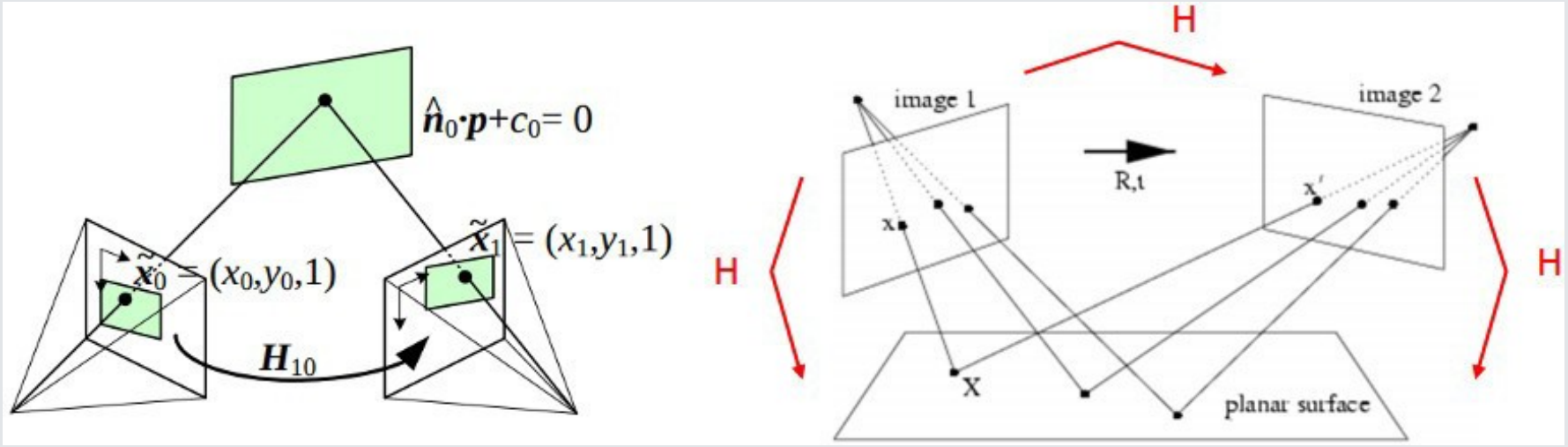
Homogeneous Cartesian

$$x_i = f\frac{x_s}{z_s}, \quad y_i = f\frac{y_s}{z_s}$$

https://math.stackexchange.com/questions/1570016/why-are-homogenous-coordinates-needed-in-image-projection

# Homography( 平面投影轉換/ 單應性)

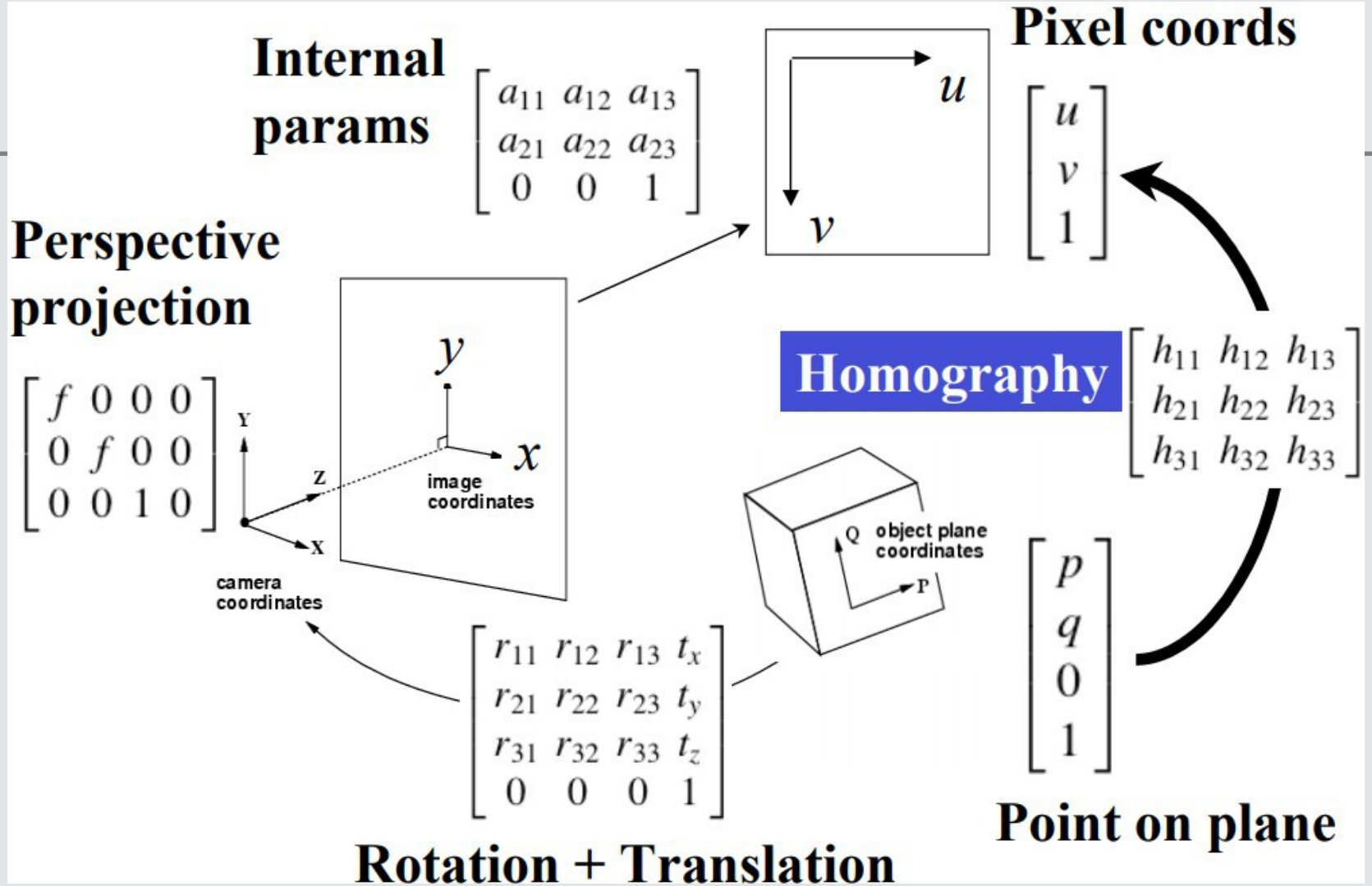
- Homography 描述了兩個平面間的變換關係，表示一個平面到另外一個平面的投影對映




$$s\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

https://docs.opencv.org/3.4/d9/dab/tutorial_homography.html

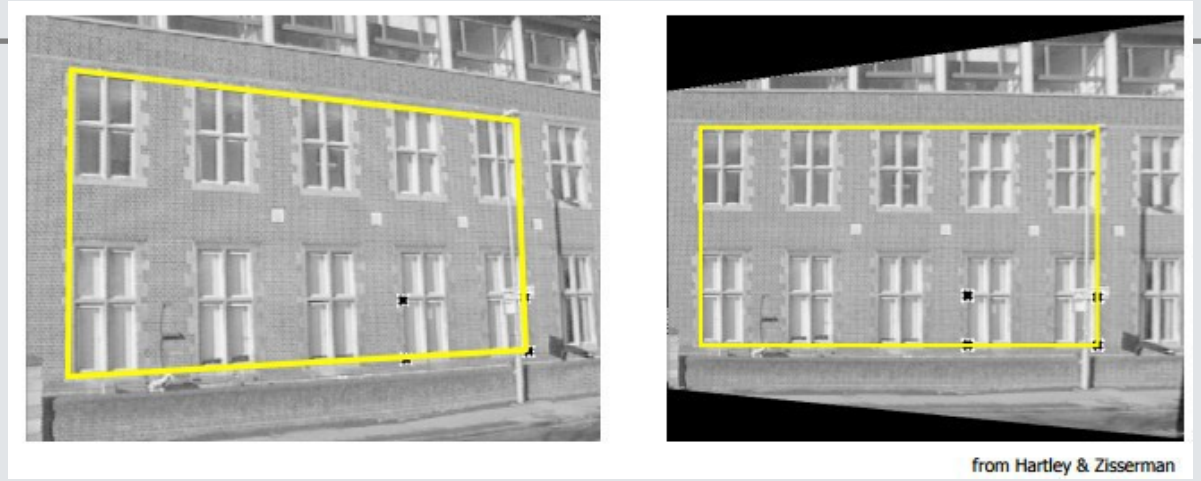
# 計算出H 可回推兩平面間的變換



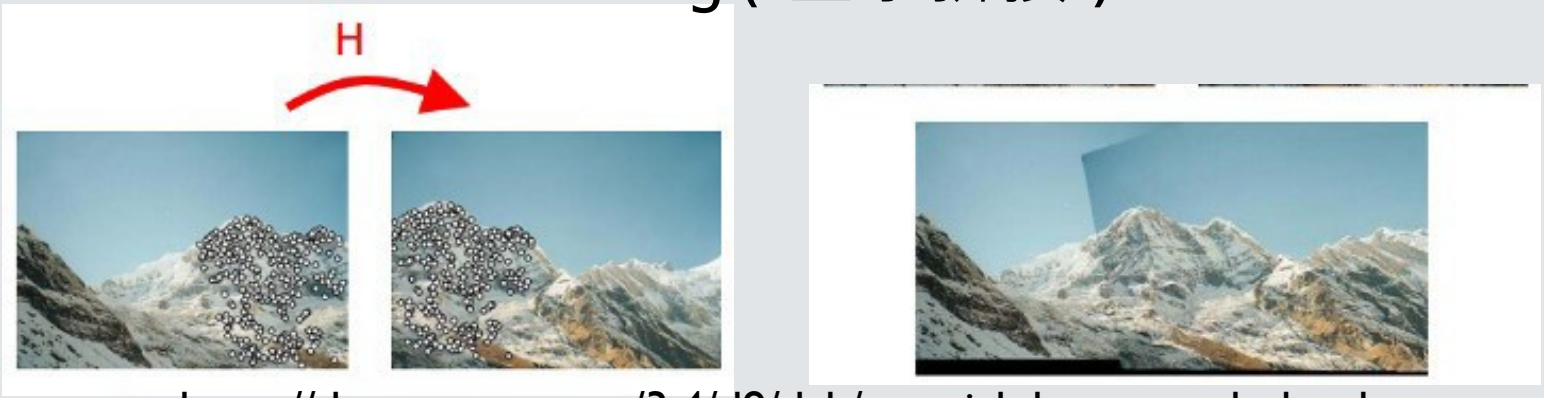

http://www.cse.psu.edu/~rtc12/CSE486/lecture16.pdf

# Homography 應用

- Perspective removal/correction( 透視去除 )



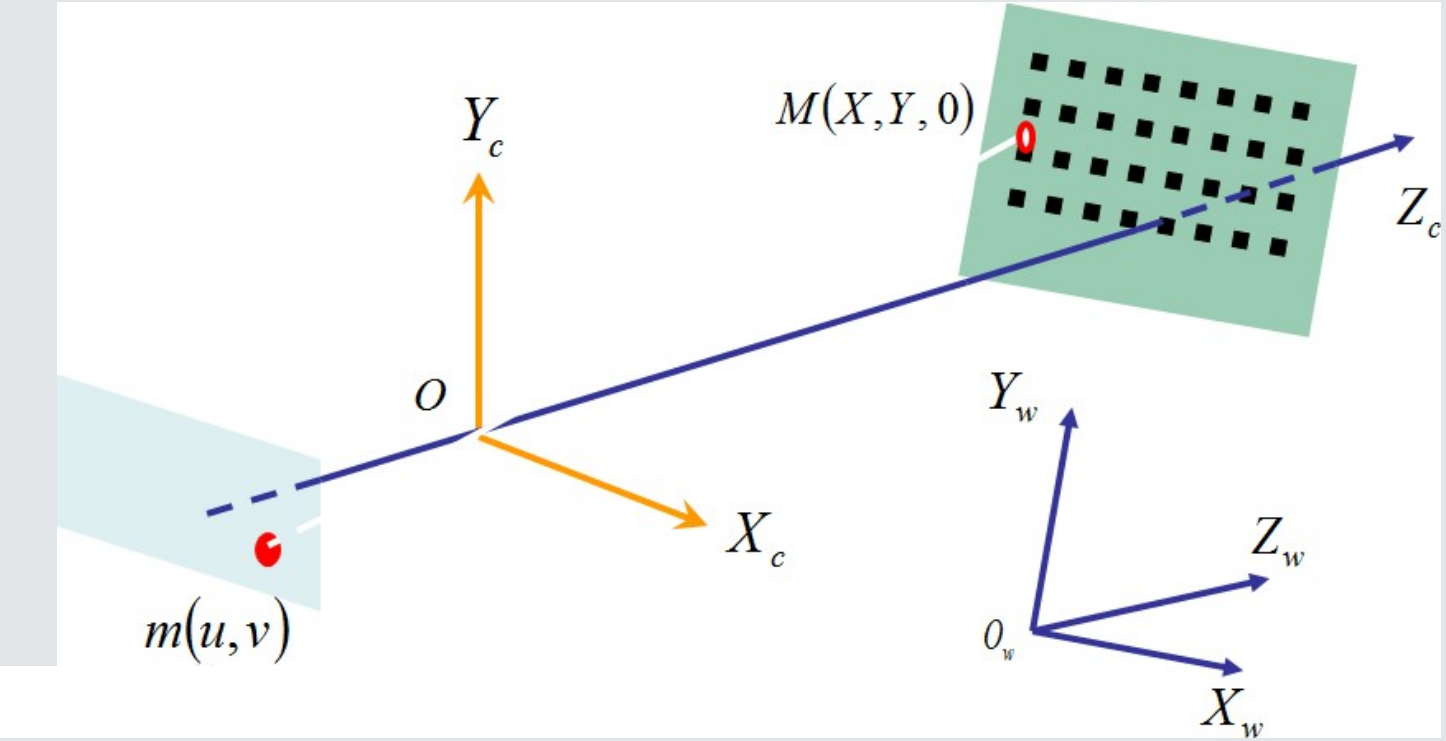

- Panorama stitching( 全景拼接 )




https://docs.opencv.org/3.4/d9/dab/tutorial_homography.html

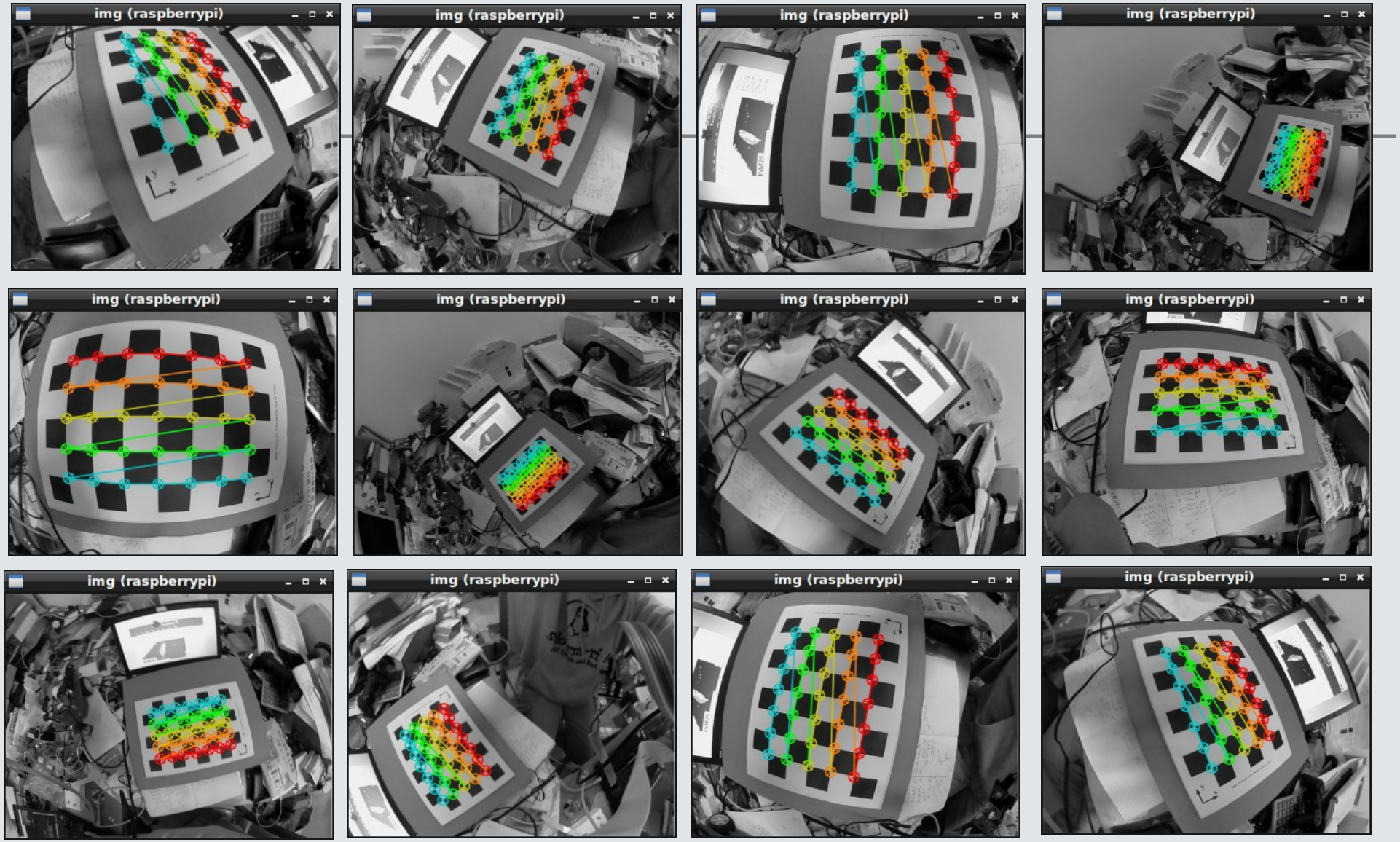
# 傳統相機校正方法(Camera Calibration)

- 張正友法 (Zhengyou Zhang)，單平面棋盤格的相機校正方法 (1998)




https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr98-71.pdf

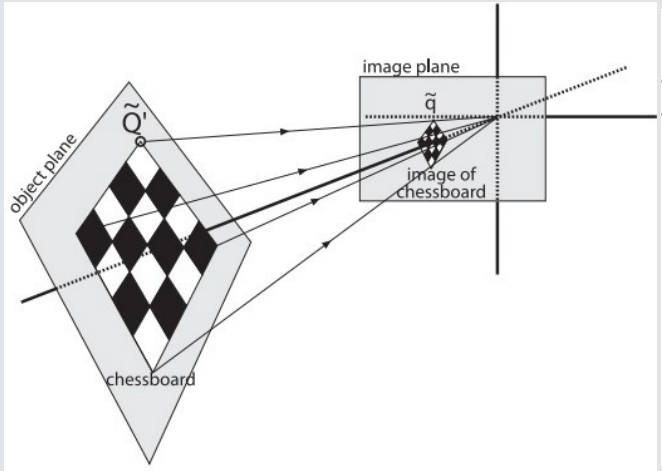
# 即時找出棋盤格角點


img (raspberrypi)
img (raspberrypi)
img (raspberrypi)
img (raspberrypi)
img (raspberrypi)
img (raspberrypi)
img (raspberrypi)
img (raspberrypi)
img (raspberrypi)
img (raspberrypi)
img (raspberrypi)
img (raspberrypi)

# 限制條件和求解

- 限制條件
  - 每個棋盤有 N 個角點，每次產生 2N 個方程式
- 求解條件
  - 找出 4 個內參和 6 個外參需幾個方程式？
  - Ans:2N>=10

Closed-form solution




$$\mathbf{B} = \mathbf{A}^{-T}\mathbf{A}^{-1} \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}$$

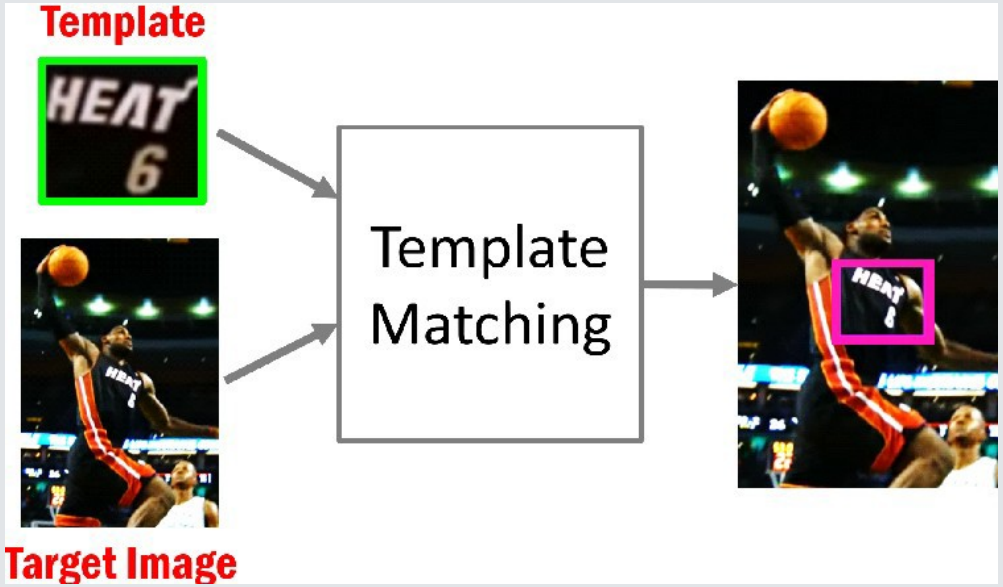
$$= \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}$$

https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr98-71.pdf

# 光相機和熱相機校正

# 樣板匹配 (Template Matching)

- 在來源圖形 (S) 中尋找目標圖形 (T)
- 以用滑動視窗比對相似度 Similarity(S,T)
- 優點：簡單，直接
- 缺點：不具有旋轉不變性，不具有尺度不變性

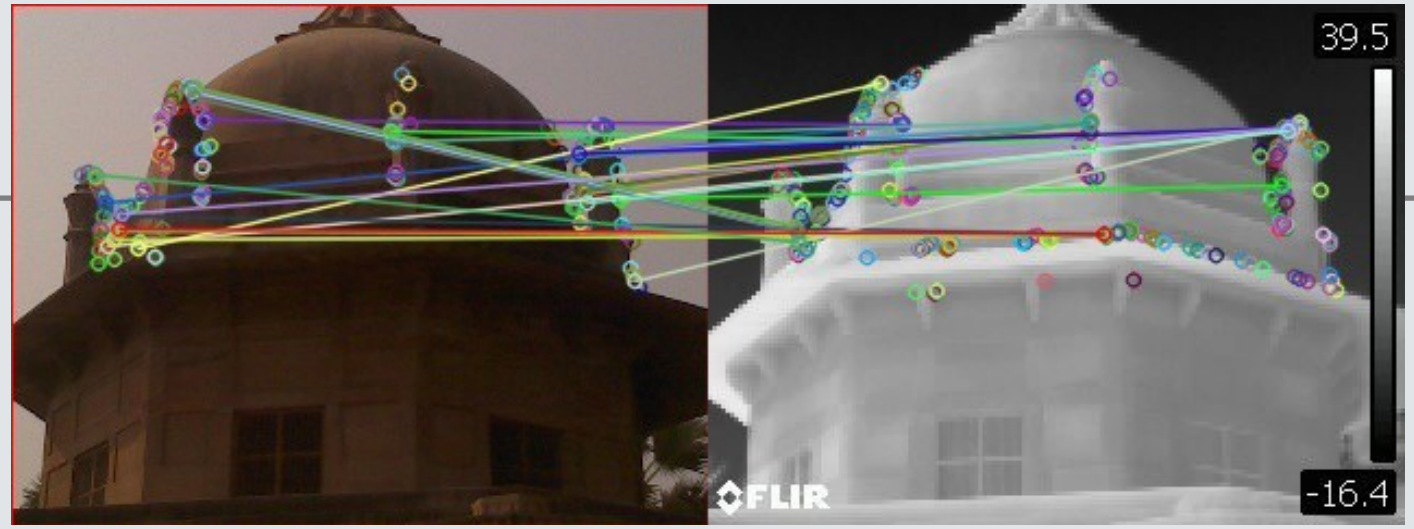



http://bit.ly/32H9BDk

# 雙相機的座標系和視角都不同


https://medium.com/@aswinvb/how-to-perform-thermal-to-visible-image-registration-c18a34894866

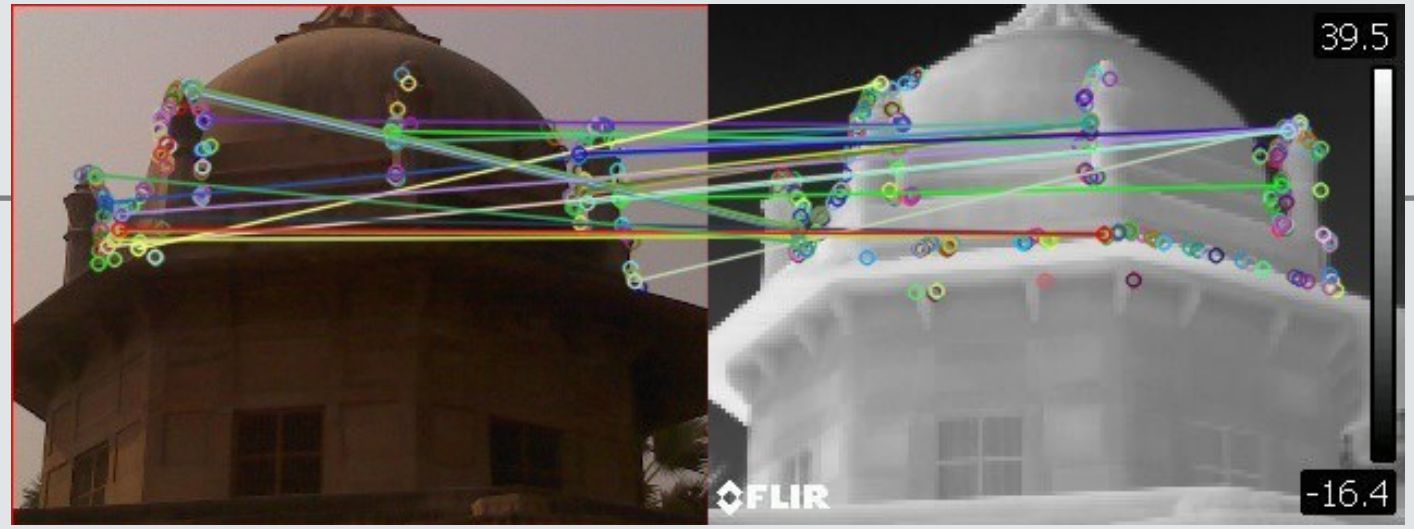
利用多尺度樣板匹配(multi-scale
- template
  match)對雙相機做校正





https://medium.com/@aswinvb/how-to-perform-thermal-to-visible-image-registration-c18a34894866

利用多尺度樣板匹配(multi-scale
- template
  match)對雙相機做校正

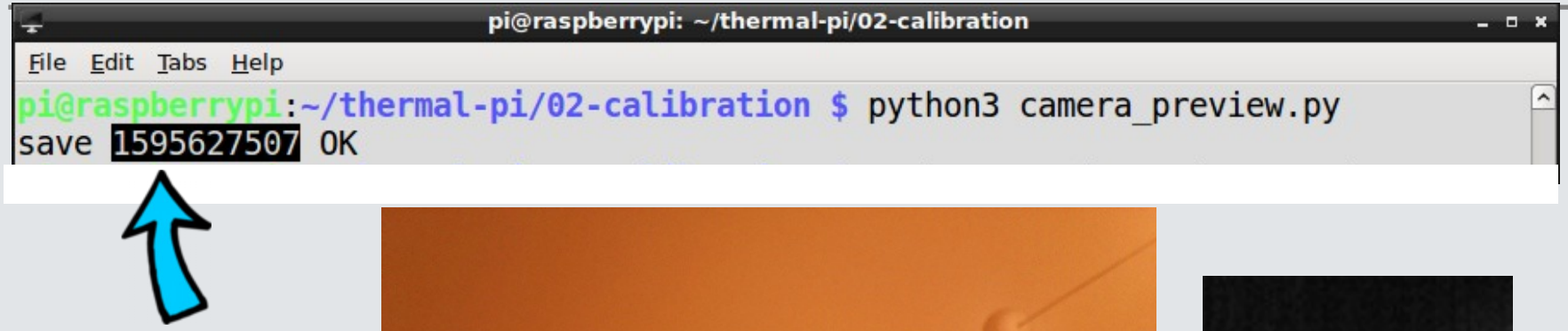




https://medium.com/@aswinvb/how-to-perform-thermal-to-visible-image-registration-c18a34894866

# 雙相機校正需要先拍照

- $ python3 camera_preview.py




自動產生的檔名

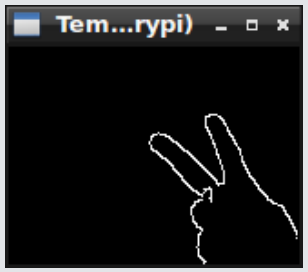

校正重點：
光影像背景要單純
熱影像物體要完整

# 拍照後根據產生檔名做校正

- $ python3 registration.py -i [FILE]

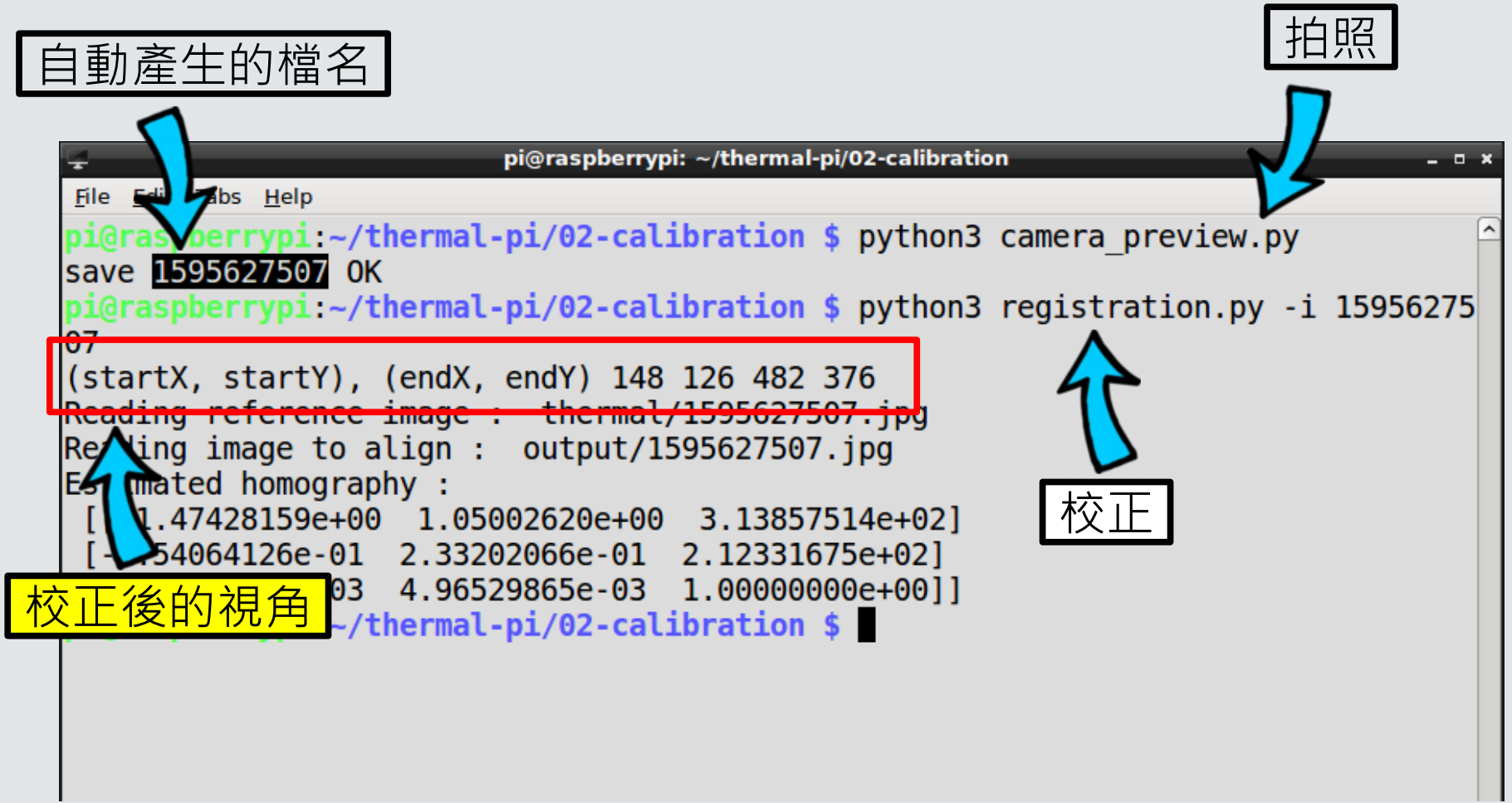
```
pi@raspberrypi:~/thermal-pi/02-calibration $ python3 registration.py -i 15956275
07
```


自動產生的檔名



# 執行結果

- `$ python3 registration.py -i [FILE]`



# 執行結果

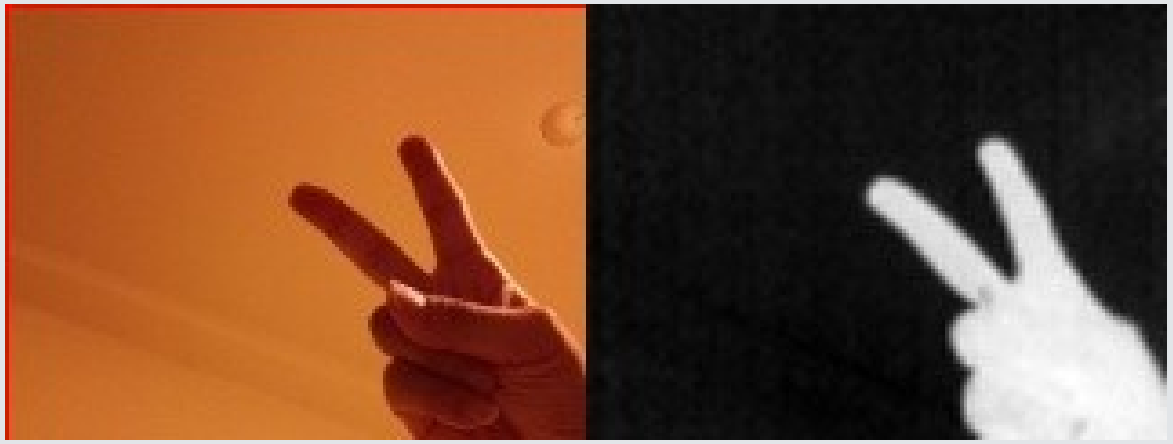
- 將剛才的執行結果修正到 `../fusion.conf`

FLIR > thermal-pi > fusion.conf

```
1   [visible]
2   win_w = 640
3   win_h = 480
4
5   [stereo]
6   startx = 286
7   starty = 224
8   endx = 714
9   endy = 546
10
11
```

# 檢查校正結果

- $ gpicview results/1595627507.jpg



每個人檔名不同

# 自動讀取校正後的視角

讀取 fusion.conf 的視角

```
config = configparser.ConfigParser()
config.read('../fusion.conf')
startX = int(config.get('stereo', 'startX'))

with Lepton() as l:
    while True:
        a, _ = l.capture()
        cv2.normalize(a, a, 0, 65535, cv2.NORM_MINMAX)
        np.right_shift(a, 8, a)
        _a = np.asarray(a, np.uint8)
        _a_rgb = cv2.cvtColor(_a, cv2.COLOR_GRAY2RGB)
        img1 = cv2.resize(_a_rgb, (160, 120), interpolation = cv2.INTER_CUBIC)

        _, img2 = cap.read()
        img2 = imutils.resize(img2, 640)
        crop_img2 = img2[startY:endY, startX:endX]
        crop_img2 = cv2.resize(crop_img2, (160, 120))
```

# DEMO
# calibrated_dual_camera.py

```
$ cd ~/FLIR/thermal-pi/02-
calibration
$ python3 calibrated_dual_camera.py
```

# DEMO
# calibrated_blend_camera.py

```
$ cd ~/FLIR/thermal-pi/02-calibration
$ python3 calibrated_blend_camera.py
```

# 先找出人臉,再計算出人臉最高溫度

# 人臉偵測與人臉識別

- Facial Detection:
  Where is the face?

- Facial Recognition:
  Who is this?



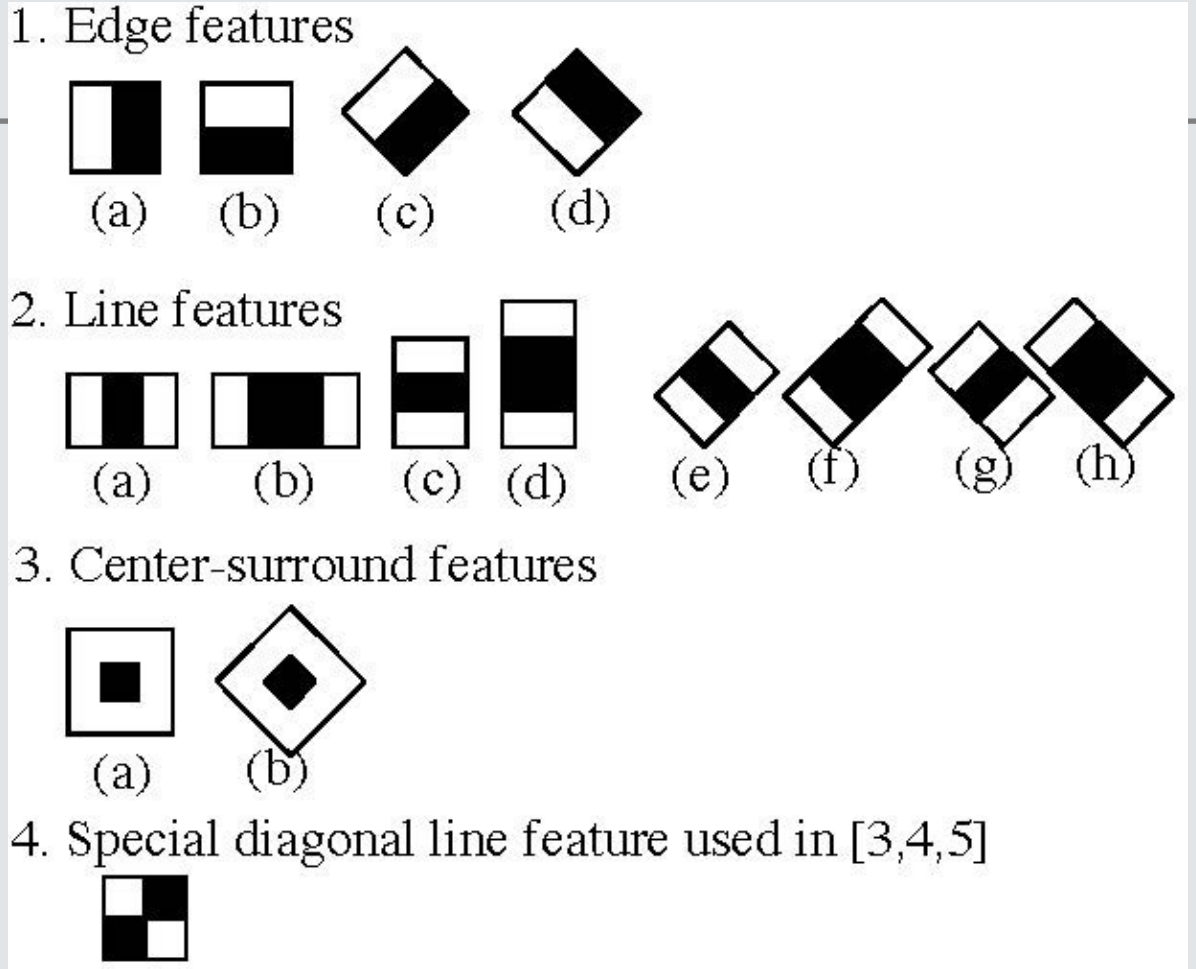
https://www.youtube.com/watch?v=sWTvK72-SPU

# Haar Feature Cascade

- 由 Viola & Jones 提出，並由 Lienhart & Maydt 改善
- 採監督式學習的類神經網路演算法，並有以下特色
  - 特徵比對 (Haar features)
  - 積分影像計算 (Integral Image)
  - 串接分類器 (Cascade)
  - 學習機制 (AdaBoost)

http://www.open-electronics.org/raspberry-pi-and-the-camera-pi-module-face-recognition-tutorial/

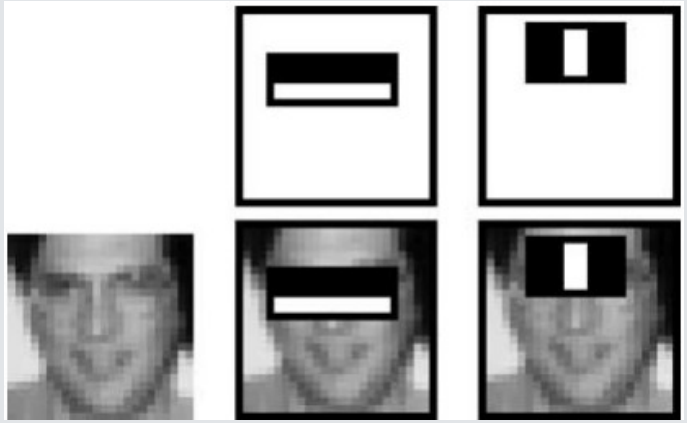
# Haar-Like Features




http://archive.cnx.org/contents/d13cd00c-cabf-402b-844a-0cbe92aaef3f@2/algorithms

# 特徵比對

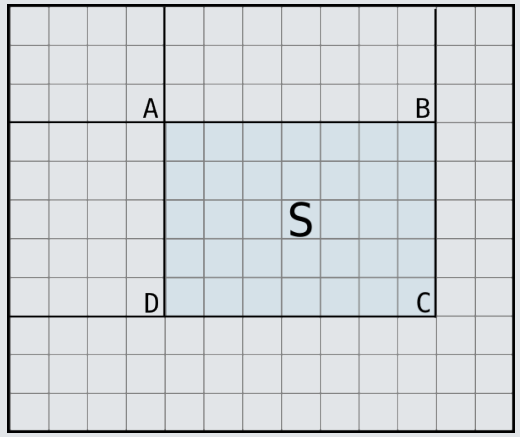
- 定義一個 24x24 的檢測窗口
- 每個特徵由左到右,由上到下滑動比對
- 計算特徵裡黑色和白色區域的灰階值
- 如果兩個區域灰階值的差大於門檻值,該特徵保留

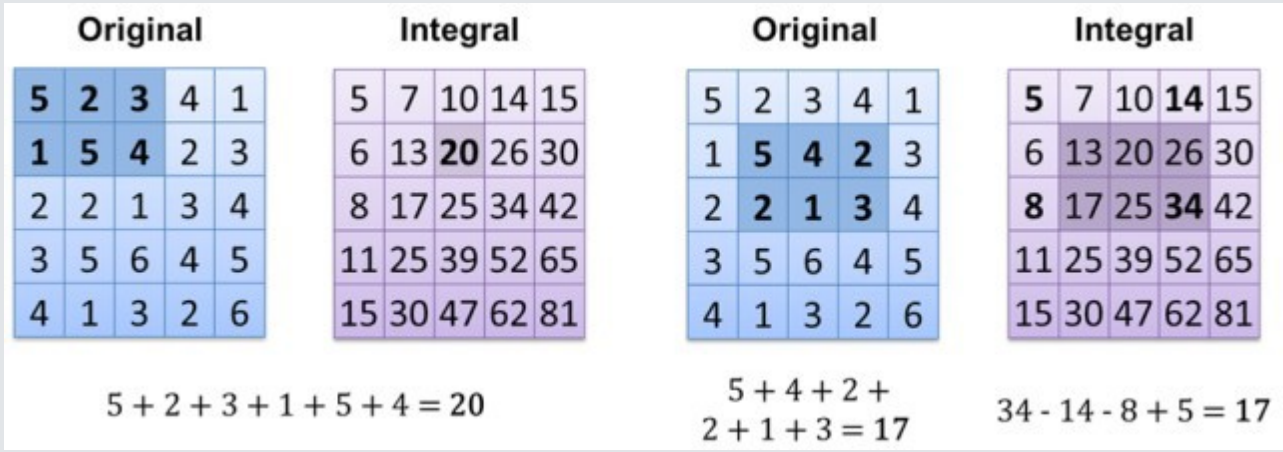


https://www.youtube.com/watch?v=sWTvK72-SPU

# 積分影像計算

- 計算灰階值可以先將灰階值的加總 ( 積分 ) 算一輪
- 任何大小的矩形灰階值加總可由四頂點的矩形灰階值加減取得




http://computervisionwithvaibhav.blogspot.tw/2015/08/viola-jones-in-nut-shell.html

# Haar Cascade

- Cascade 是一連串 Haar-like features 的組合所形成的分類器 (classifier)
- Haar Cascade = Classifier

https://www.youtube.com/watch?v=sWTvK72-SPU

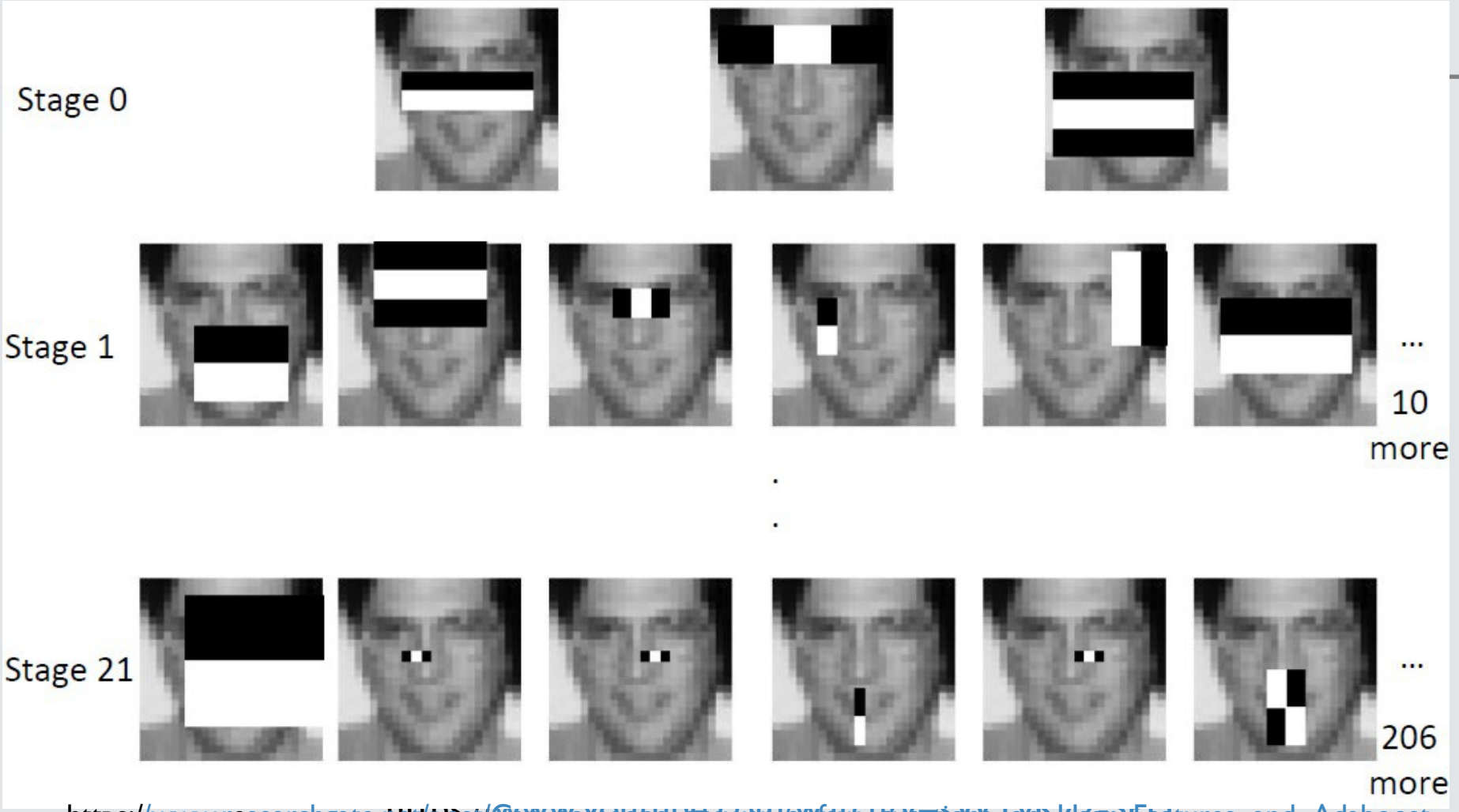
# 弱分類器

- Feature:  and 
- Classifier:  

- 單一的分類器準確度不夠，因此也稱為是弱分類器

https://www.youtube.com/watch?v=sWTvK72-SPU

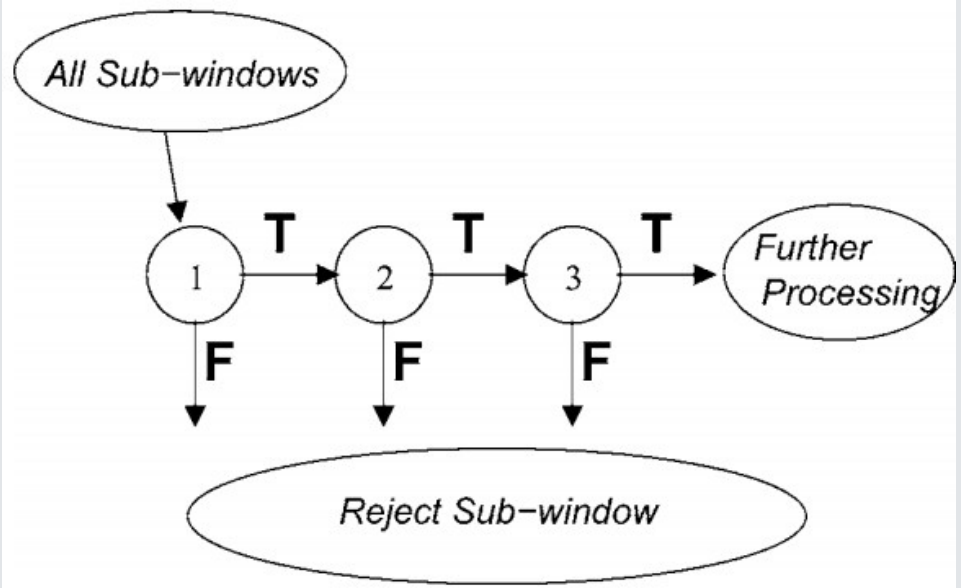
# 串接多個弱分類器




https://www.researchgate.net/post/Can_someone_answer_a_few_questions_on_Haar_Features_and_Adaboost

# 串接弱分類器的好處

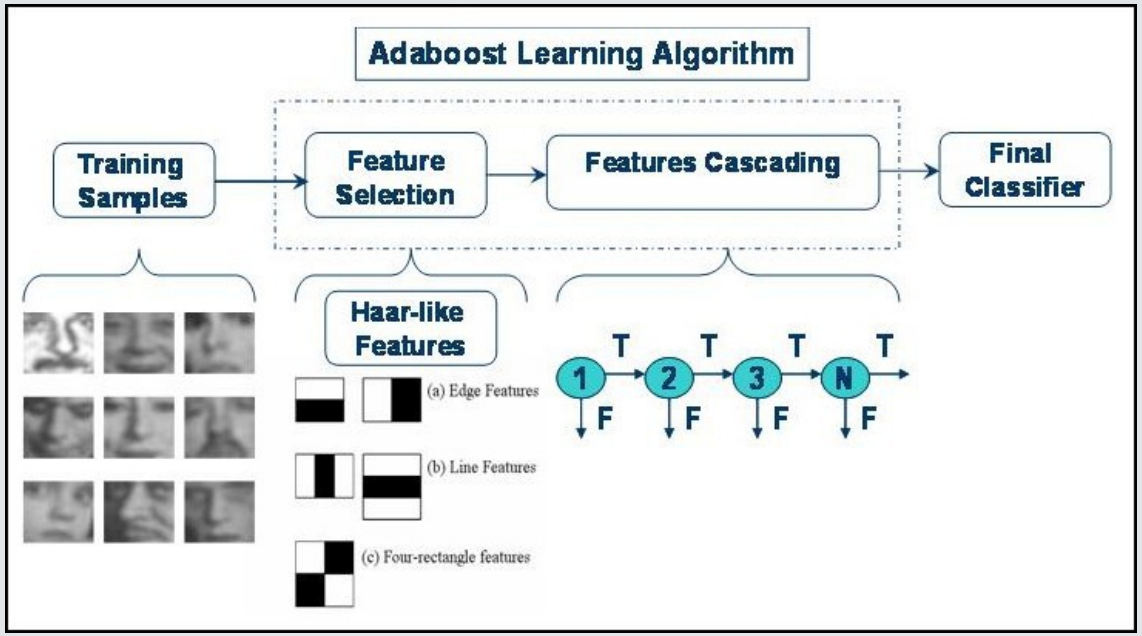
- 單一分類器的準確度雖然只略大於50%，但如果同時通過多個弱分類器，其準確度將會大副提昇
- 使用弱分類器的好處是可以將未達到門檻值的特徵(feature)在早期就先去除掉




https://www.youtube.com/watch?v=sWTvK72-SPU

# AdaBoost(Adaptive Boosting)

- Adaboost 在學習階段可以組合效果好的分類器，並根據監督式的學習過程調整不同分類器的權重，用來建立適合的偵測模型

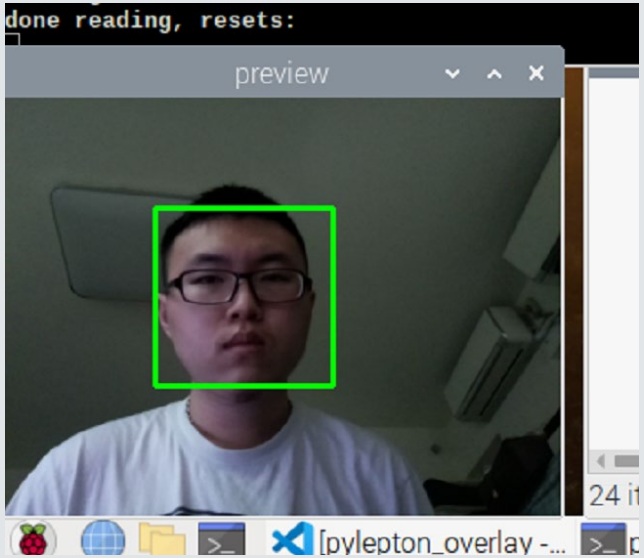



https://www.youtube.com/watch?v=sWTvK72-SPU

# 讀取Camera 並辨識

```
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize=(30, 30),
        flags=cv2.cv.CV_HAAR_SCALE_IMAGE
    )

    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

# DEMO
# opencv_face_detect.py




```
$ cd ~/FLIR/thermal-pi/03-fusion
$ python3 opencv_face_detect.py haarcascade_frontalface_default.xml
```

# 自製雙相機熱像儀

- 先找出人臉,再算出人臉平均溫度(或最大溫度)