Photonic Workshop
2019 SUMMER 實習報告

鄭泊聲

摘要

本次實習我主要的工作內容是完成一台採用全幅相機作為感光元件的 OM。過程中主要分為硬體與軟體兩個部分。軟體在羅詔元博士的幫助下，首先學會使用 LabVIEW，接著學習 Mightex CCD 的 SDK 開發方式，最後進入關鍵的 Canon LabVIEW 相機操控程式開發，過程中解決了 labview 與 c 語言 pointer 的記憶體資料調用問題，並且額外的學會了自己建立動態函式庫 dll，最後成功在軟體部分完成能夠操控、拍攝、傳檔的 Canon LabVIEW 相機操控程式。

硬體部分，在張玉明老師與實習夥伴張振淙的幫忙下，我們成功組合出一台專門為全片幅感光元件所設計的 OM，能夠看見非常大的視野範圍。過程中花了相當多時間在進行照光光源的調整，經過共阨焦設計、針孔、毛玻璃、成像透鏡等等元件的實驗後，我們發現採用兩吋光學元件系統的 OM 照明系統，竟然沒有比一寸光學元件系統的照明系統提供更大的照明範圍，這是值得深究的議題。

目錄

一、 全幅機的 OM 應用

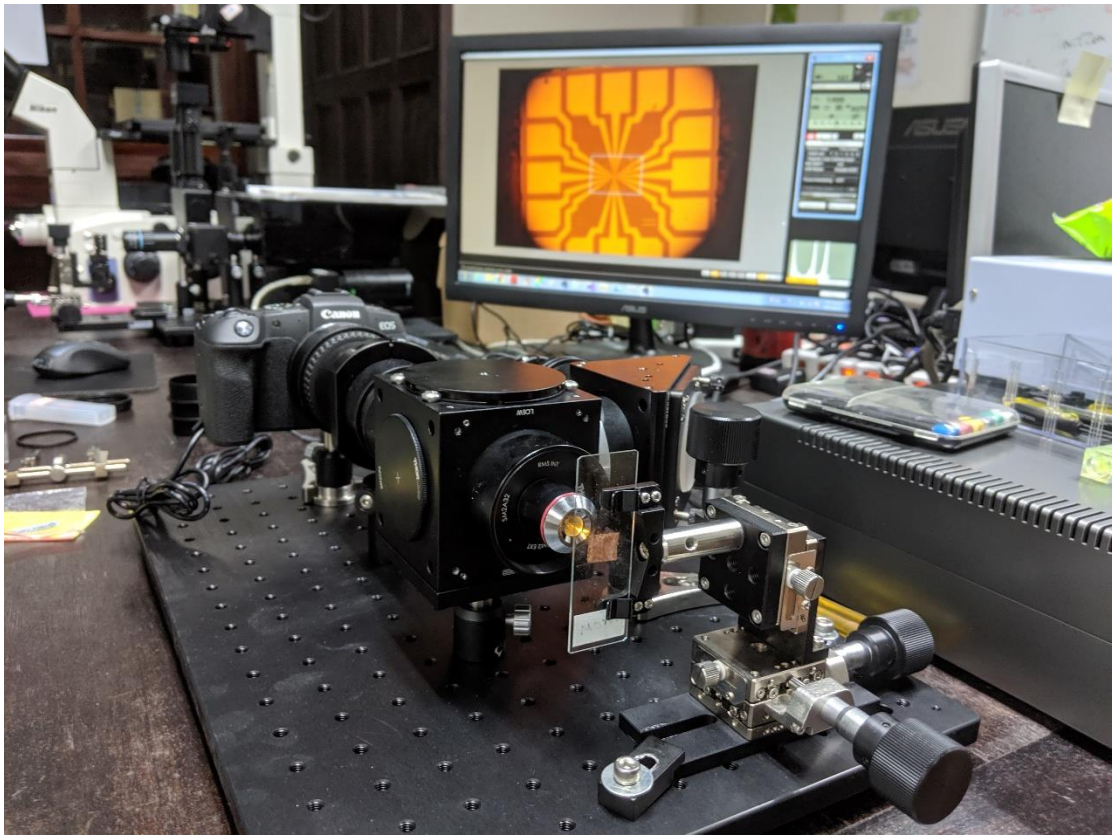
全幅機 OM 系統概述

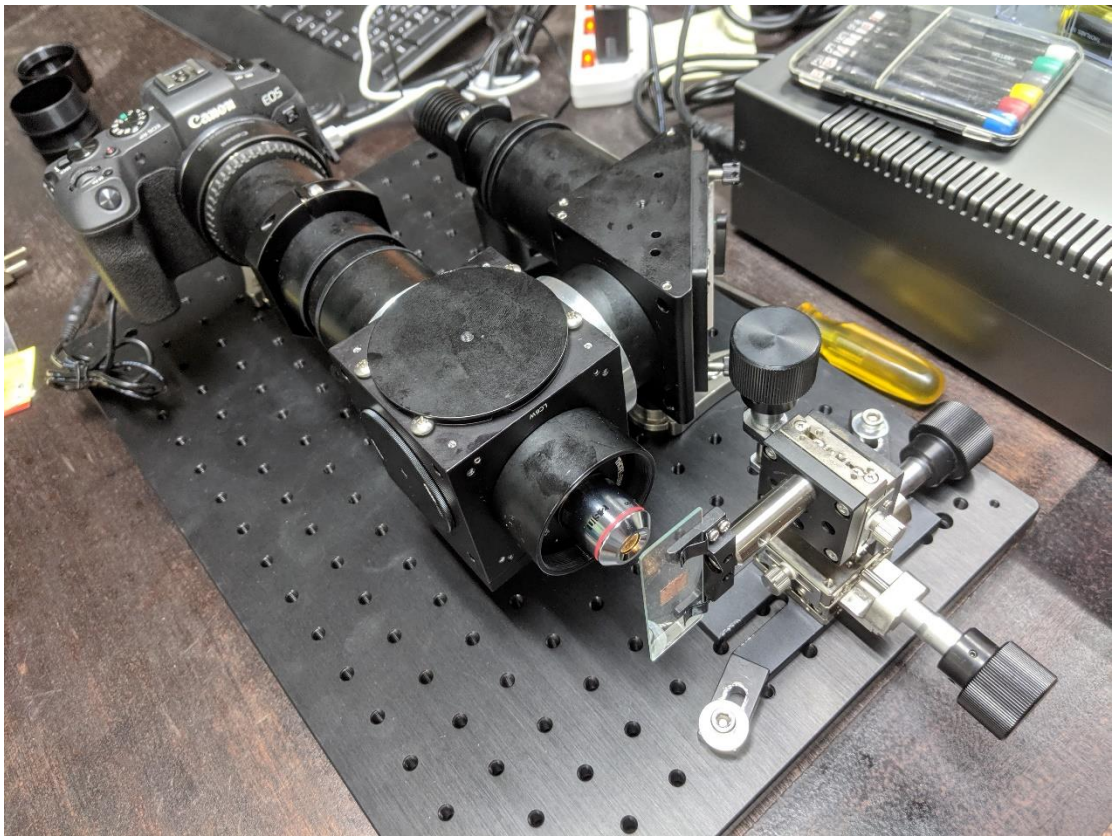
過去採用科學用 CCD 作為 OM 的感光元件，但其價格高昂，且實際能接收到的成像範圍極小，限制了 OM 的 FOV。現今消費型的可換鏡頭相機，已擁有非常高的像素數、全片幅的大感光元件、能夠用於自行開發軟體的 SDK，因此我們想嘗試使用消費型的商業相機來取代科學用 CCD。

本系統中央是一組四面轉接盒，其內有一 50/50 分光鏡，面對分光鏡反射鍍膜面的兩接口(它們彼此垂直)，分別接上光源與物鏡，而在物鏡接口對面，與其同軸的接口，則接上我們所採用的全幅相機。以上，轉接盒的四面接口總共使用了三面，第四面將其封閉。

光源進入轉接盒後，首先遇到分光鏡的反射鍍膜面，50%的光線會反射至物鏡方向，透過物鏡照射到被觀測物。被測物受到照光後，反射出的光線穿過物鏡，遇到分光鏡(同樣是其反射鍍膜面)，50%的光線穿透分光鏡，射向相機。

在物鏡端，除了物鏡外，沒有其他光學元件，物鏡外側即為載物台。在相機端，有一凸透鏡，將被測物所反射(並透過物鏡、分光鏡)之光線，匯聚成像於相機之感光元件上。這片凸透鏡決定最後成像在感光元件上之像的大小，若所成之像小於全片福感光元件尺寸，則我們能完整看到整個物鏡的視野範圍，但浪費了許多全幅感光元件上沒有被光照射到的像素；若所成之像大於全幅感光元件尺寸，則我們只能看到整個像中的一部份，亦即無法觀測到物鏡的整個視野，但不會浪費任何感光元件上的像素。最終我們選取了焦距 20 公分的凸透鏡。在光源端，我們採用了單顆 LED 光源，並加入數個不同的光學元件，來達到照明範圍足夠、亮度均勻的光源品質。最後共使用了一個 100um 的針孔、一片毛玻璃、一個直角轉接盒、一個可移動的透鏡(調整 LED 光源的照光範圍及毛利)及一片固定的凸透鏡。所有這些元件的挑選都對最終的成像品質有所影響，以下一一討論每個元件在本系統中造成的變化，多數影像(除非特別標示)皆是搭配 Olympus 4x 物鏡拍攝。


Canon
EOS


Canon
EOS

FOV

在最佳化顯微鏡視野時，我們希望能利用到整個全片幅感光元件，同時又能看到物鏡的最大視野。意即，感光元件之矩形若正好是像之內接矩形，應是能夠照滿整個全幅感光元件的最大視野。像越大，感光元件尺寸佔像的比例就越小，實際視野也就較小。挑選合適的凸透鏡焦距，成出大小適中的像，是決定視野的關鍵因素。

這個段落主要討論放在相機前，將由物鏡而來的光線成像於感光元件上的凸透鏡。首先，我們使用焦距 7 公分的凸透鏡，觀測一把鐵尺，並於系統外以 LED 小檯燈從旁照射鐵尺，並未使用系統的單顆 LED 光源。這時由於 7 公分焦距之凸透鏡所成的像很小，只佔感光元件的一部份，因此我們能完整看到物鏡的整個視野。在同樣的條件下，我們改以 20 公分之凸透鏡成像，並由影像中鐵尺的刻度來判別兩者視野的差異。

經過簡單的計算後，繪製出圖三，發現若要達成前述「內接矩形」之條件，應選用 18 公分之凸透鏡，正好是 Olympus 原廠所建議之 tube lens 焦距。不過在現實應用中，我們發現物鏡視野邊緣的影像品質已不堪使用，因此最後決定續用 20 公分焦距之凸透鏡，雖視野稍小，但也正好是較堪用的影像範圍。

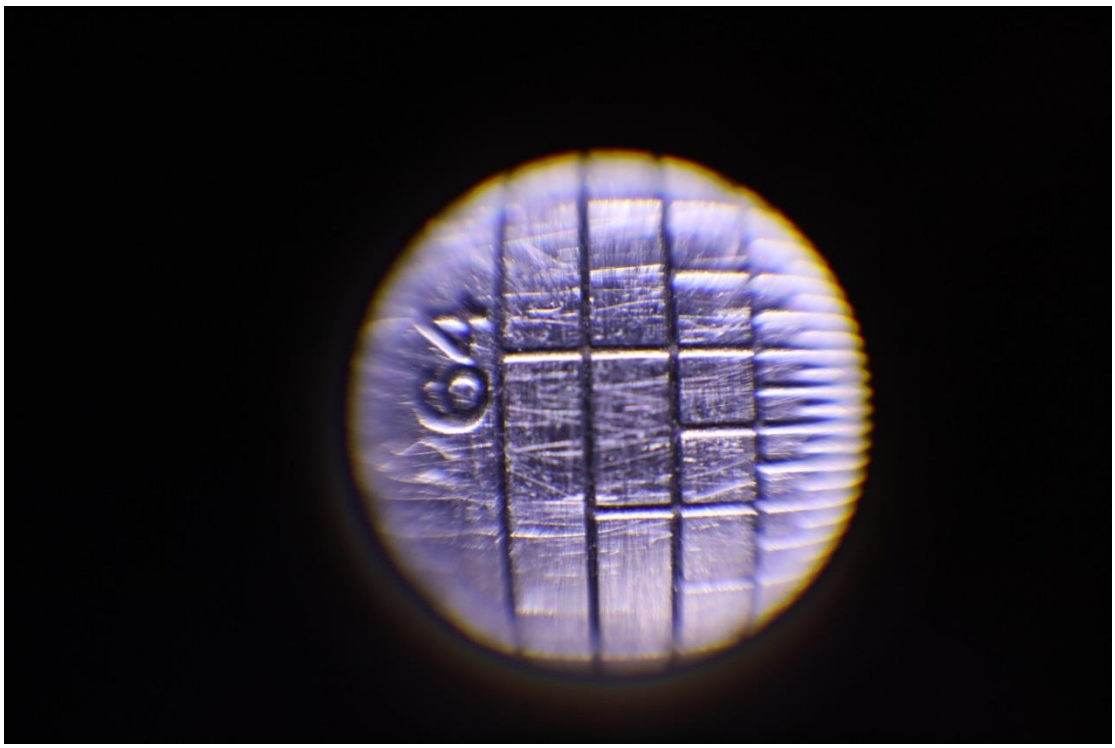


Figure 1 : F = 7cm、鐵尺、外加光源

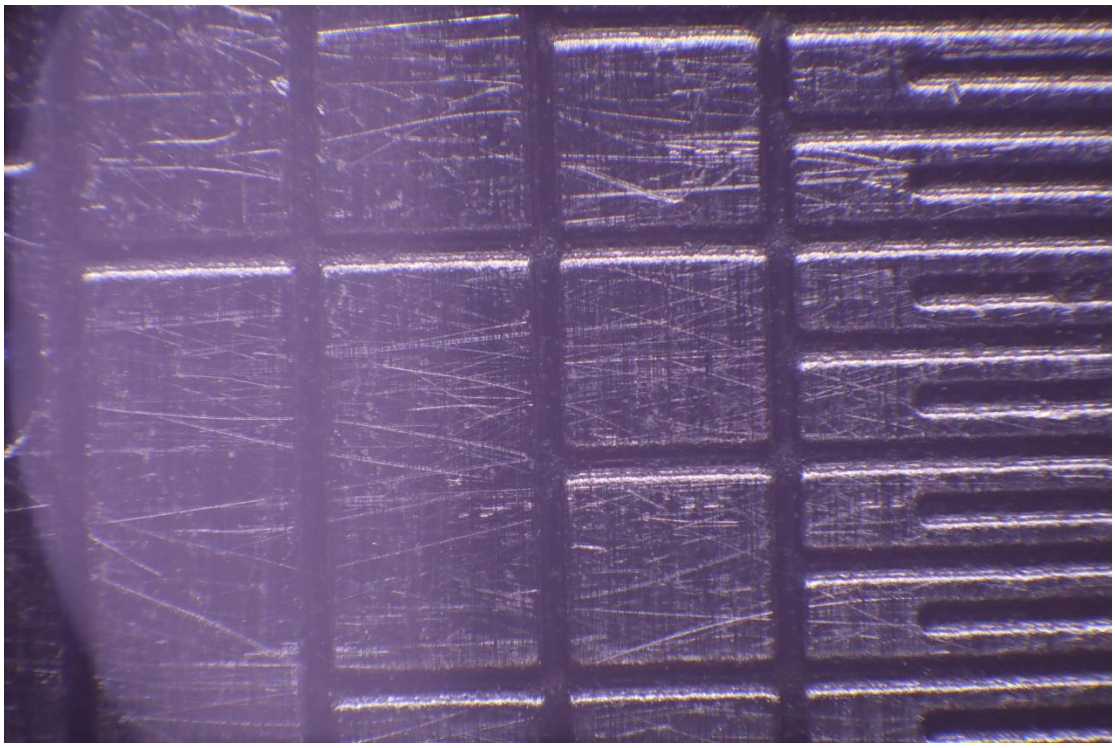


Figure 2 : F = 20cm、鐵尺、外加光源

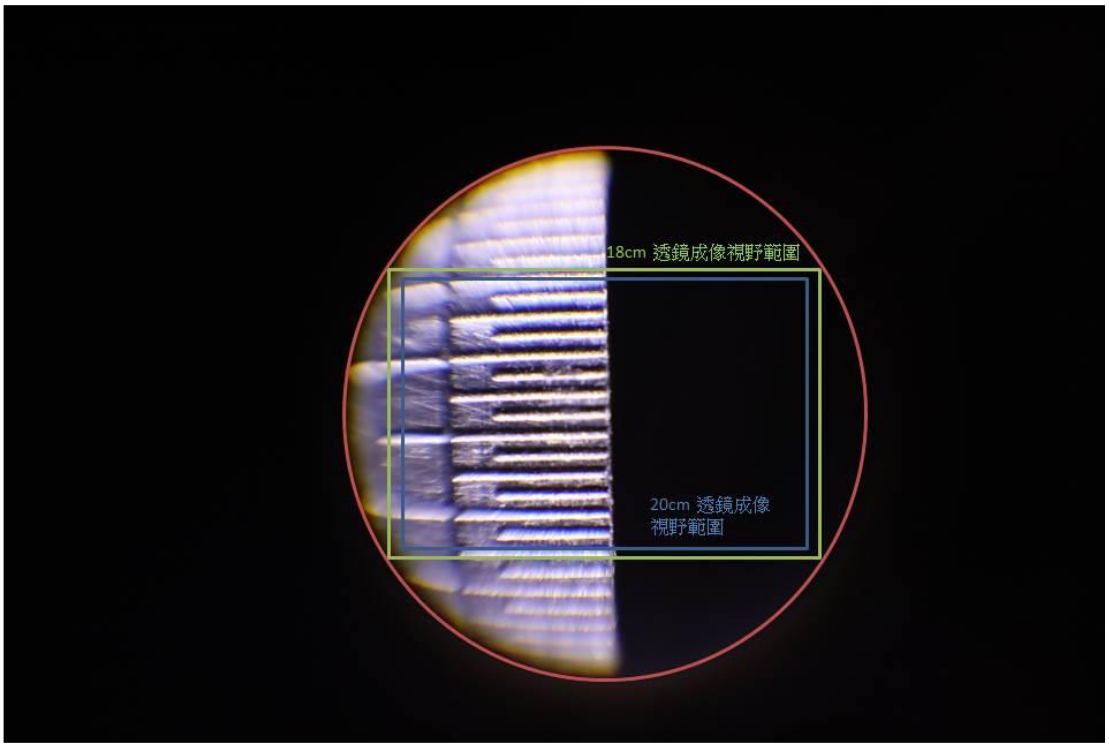


Figure 3 : 視野與焦距

Pinhole & 直角轉接

在視野的調教完成之後，我們將外部光源移除，使用系統內部的單顆 LED 光源，並觀測測試片。在直接將 LED 光源接上四面轉接盒時，呈現的影像大致如圖四，照明範圍嚴重受到光源管徑的限制，就算調整 LED 前可滑動的透鏡，使其照明範圍達到最大，依舊無法完整的照明整個視野範圍。

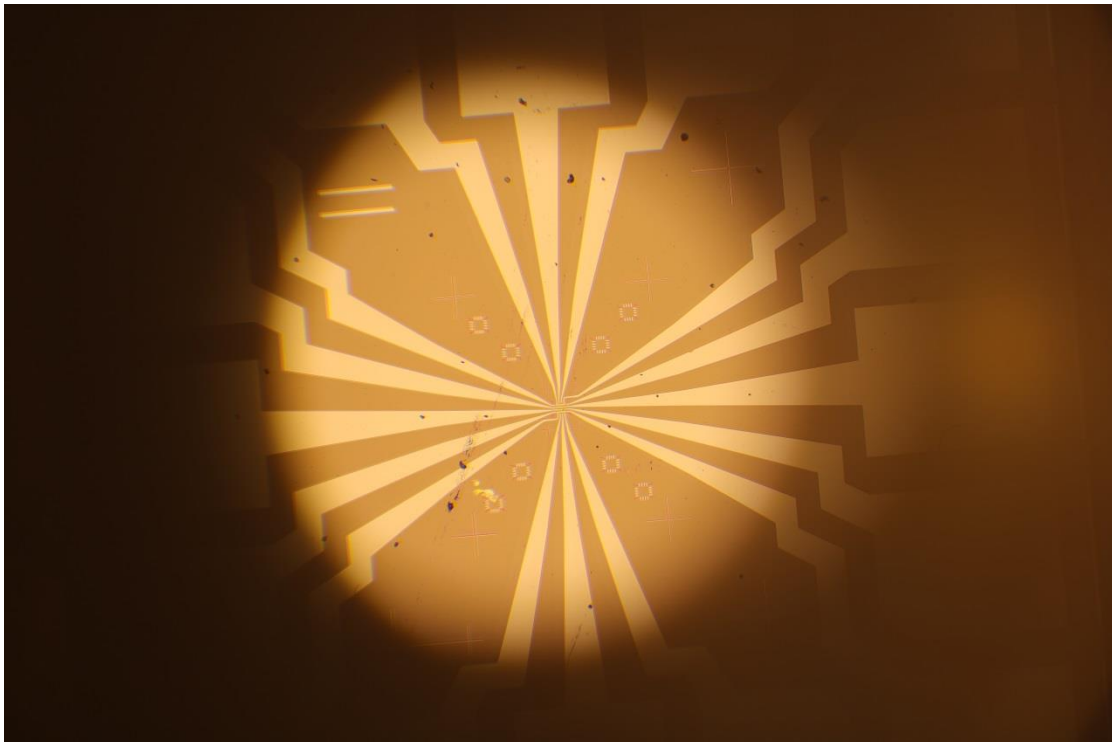


Figure 4 : 沒有針孔、沒有直角轉接

為了讓光源穩定固定在系統上，我們加上了直角轉接盒，照光範圍與原初沒有差異，依然無法照滿。

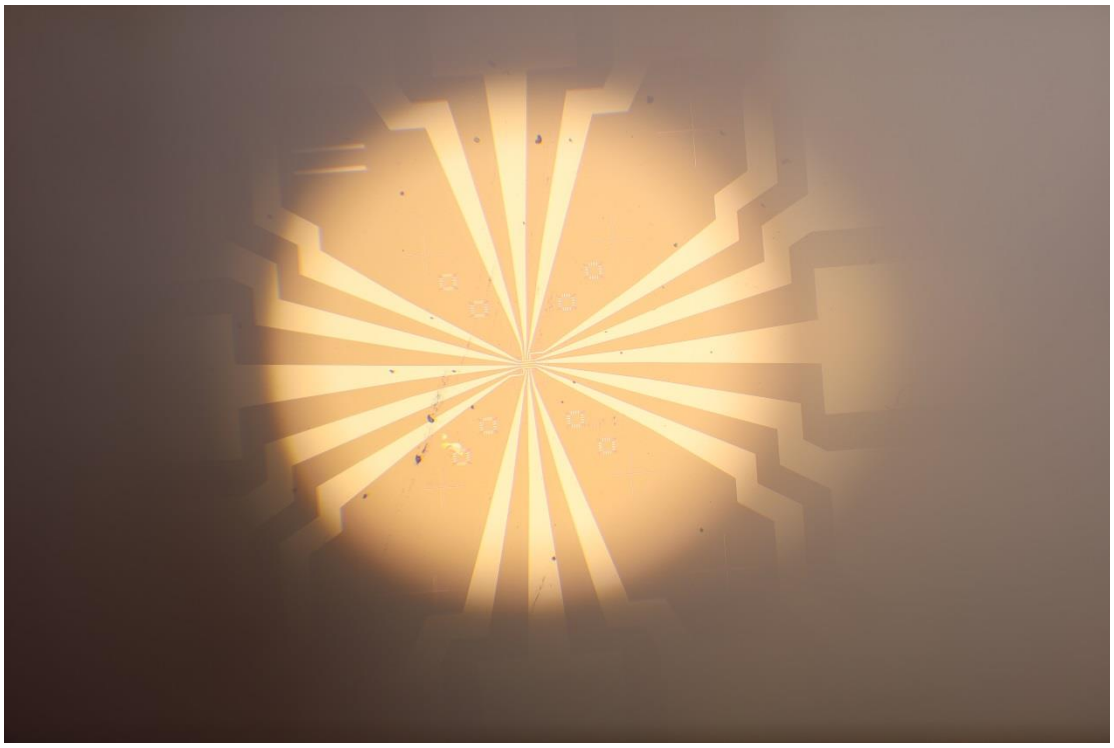


Figure 5 : 沒有針孔

沒有針孔時，若對焦不準確，甚至會發生 LED 本身成像至感光元件上的情形，如圖六。

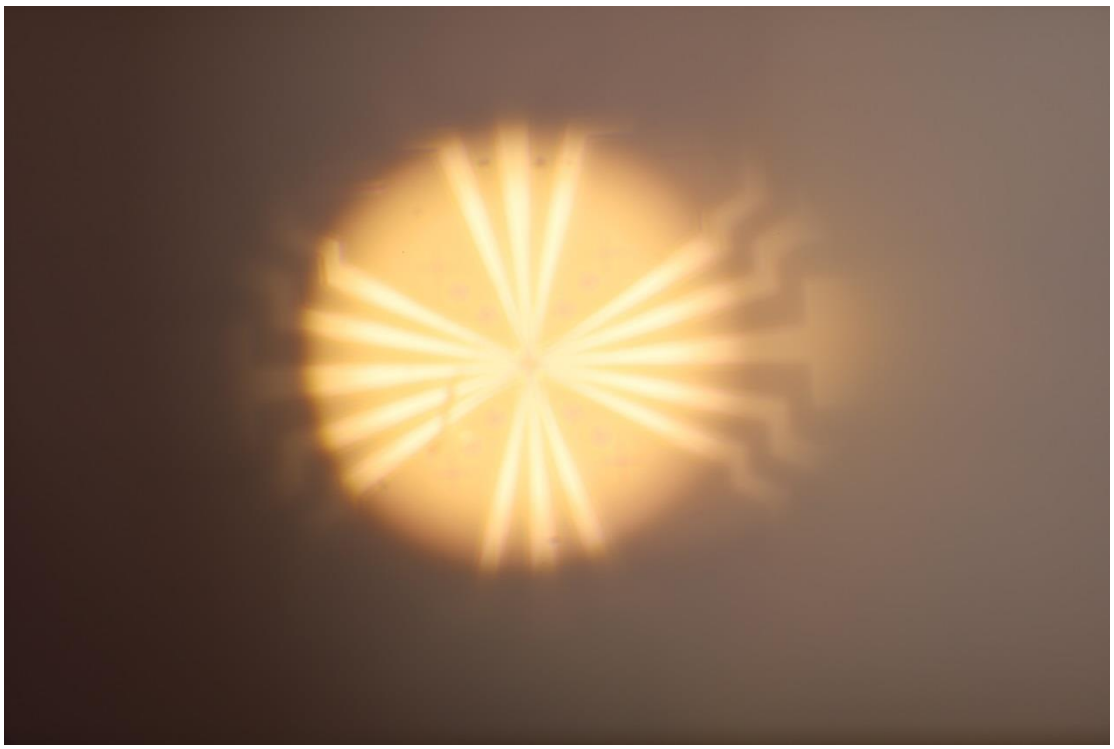


Figure 6 : 中央的 X 形圖案為 LED 顆粒的紋路

加入針孔後，將 LED 重塑為點光源，經過 LED 前可滑動的透鏡之後，最大的照明範圍足以覆蓋視野的短邊，且影像品質也有增進，但仍然不足以照滿視野。

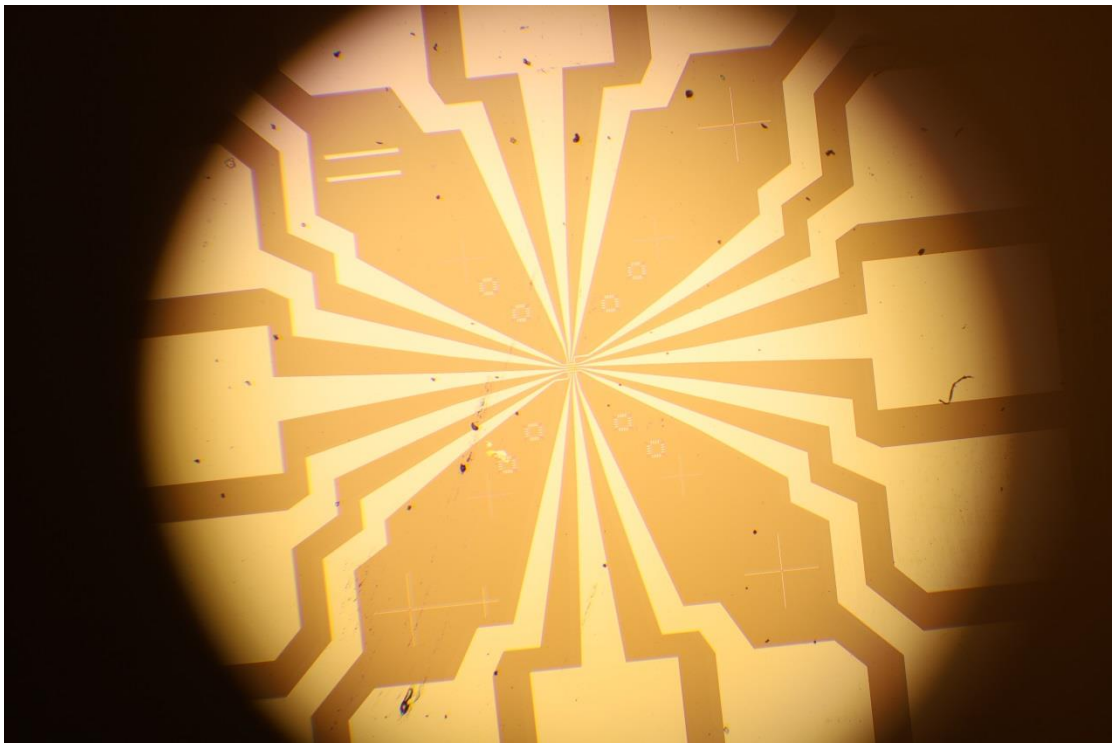


Figure 7 : 有針孔、有直角轉接

Condenser

我們在光源的直角轉接器之後，四面轉接盒之前，再加上一片凸透鏡，以將光源 tube 的光準確匯聚於被測物上。從圖八能夠看出，加上這片凸透鏡後，光線能夠完整照明整個視野。另外，我們在四面轉接盒的上方也加上蓋子，從圖八與圖九比較能發現，加上蓋子後，畫面中的色散明顯降低。

至此，整個系統已經能善用整片全幅感光元件來擷取影像。也能搭配不同倍率的物鏡使用，圖十和圖十一分別為 10x 及 20x 被物鏡的影像，但裝上物鏡切換鼻輪後，照光範圍會被限制，因此出現四周暗角。也能看見畫面中有不少汙點，影像品質還有進步空間。

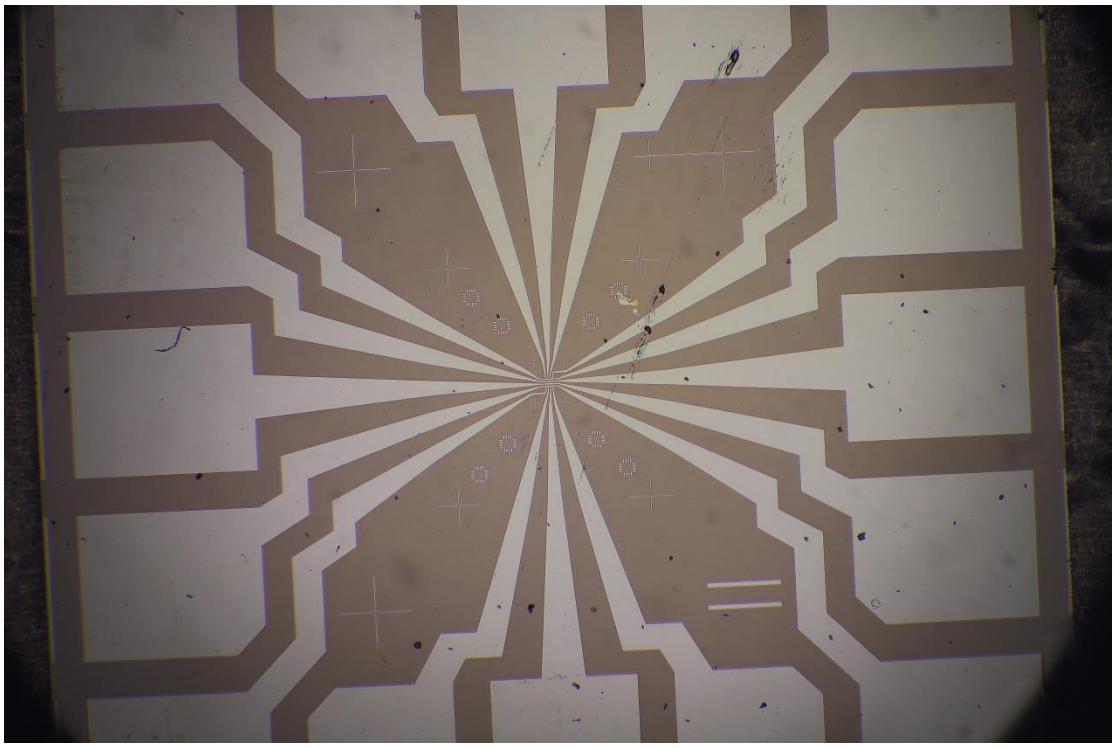


Figure 8 : 有 condenser、轉接盒加蓋、4x

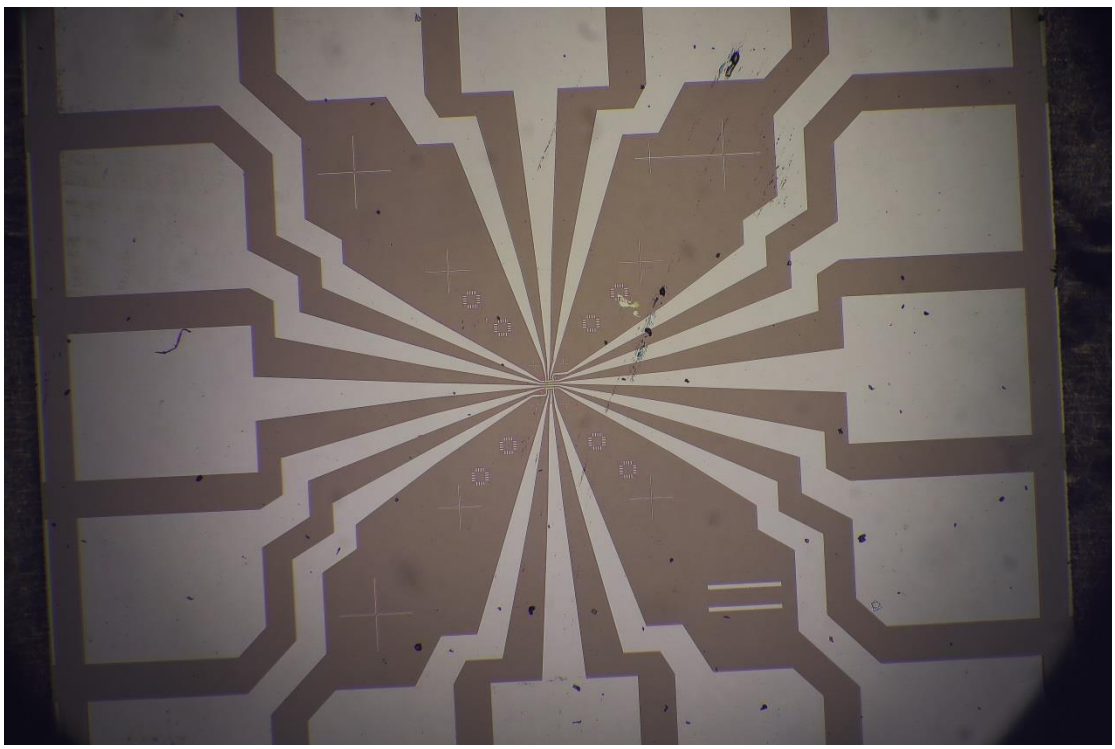


Figure 9 : 有 condenser、轉接盒未加蓋、4x

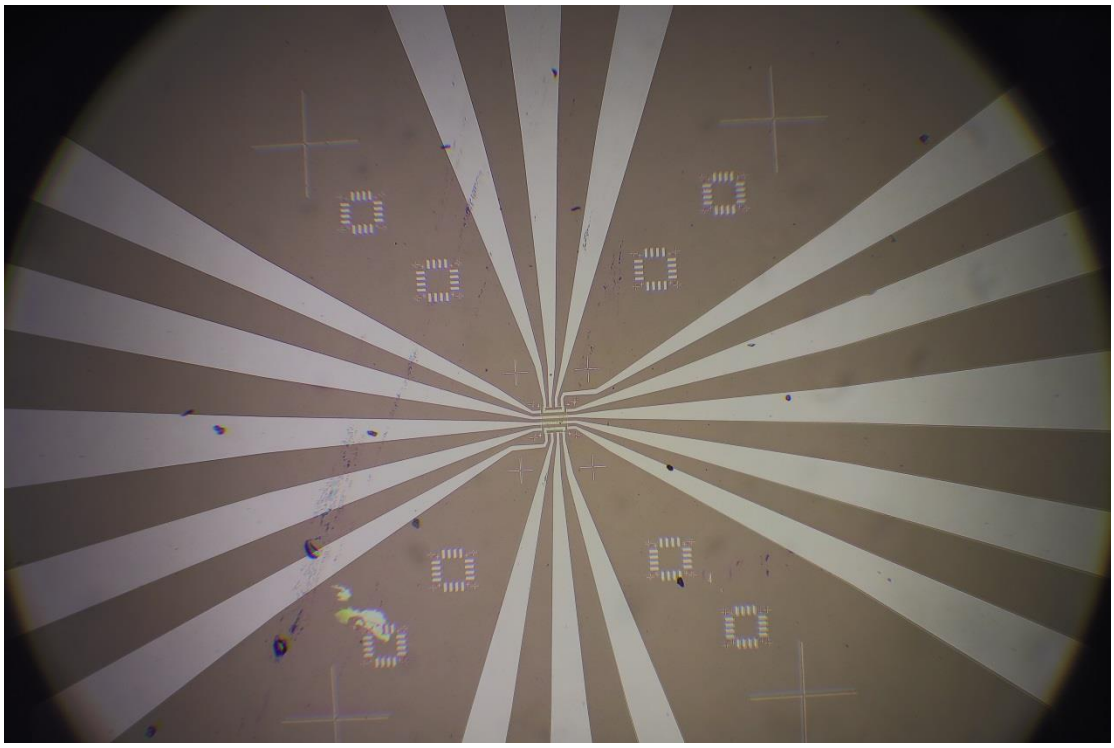


Figure 10： 有 condenser、轉接盒加蓋、10x

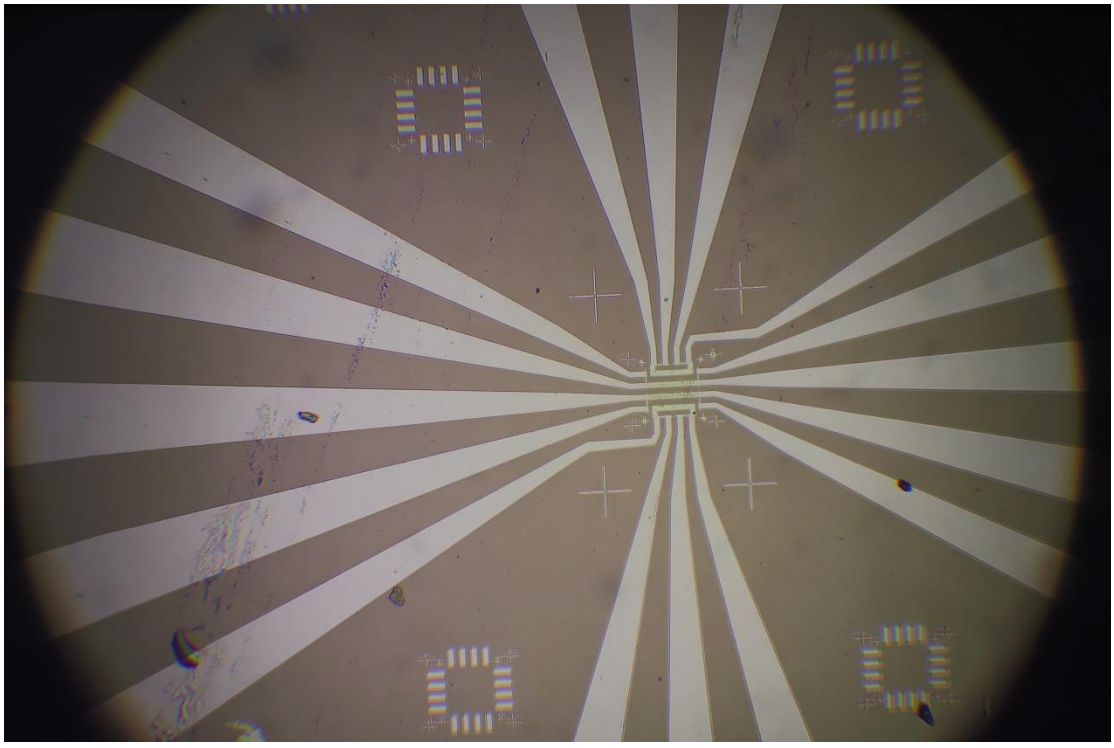


Figure 11： 有 condenser、轉接盒加蓋、20x

Diffuser

在目前的光源配置中，整個光源系統共有 LED、pinhole、毛玻璃、可滑透鏡、直角轉接、condenser 凸透鏡共六個光學元件。我們以純矽片做為測試片，實驗毛玻璃位置帶來的影響。圖十二是將毛玻璃放置在 pinhole 之前(毛玻璃離 LED 較近)，圖十三是將毛玻璃放置在 pinhole 之外，可以明顯看到將毛玻璃放置在 pinhole 之外，能夠消除光源中原本的汙點，更加提升照光品質。同時，毛玻璃也能模糊 LED 本身的像，進一步解決圖六的情形。

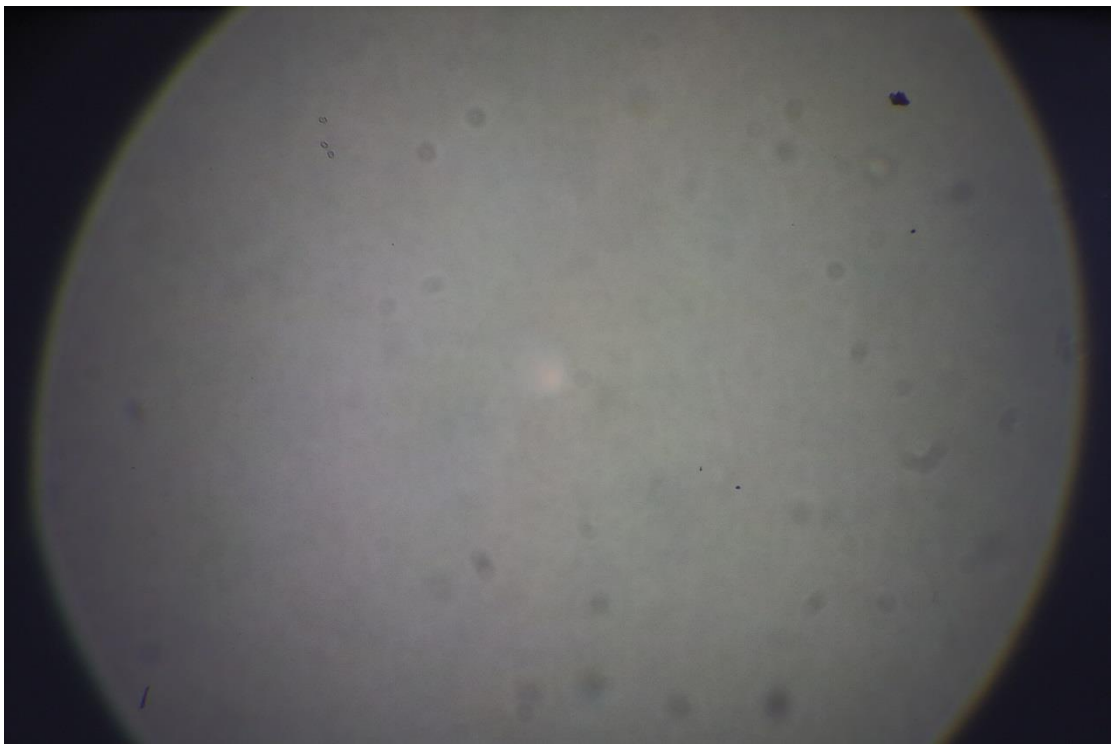


Figure 12 : sillicon sample-diffuser inside

Figure 13 : sillicon sample-diffuser outside

Pinhole size

針孔的尺寸對於照光的均勻度也有影響，在圖十四與圖十五的比較中，可以明顯看出，採用 pinhole 尺寸較大的圖十四，其畫面中央有較明顯的一圈亮圓，而周圍則較暗。相對的，pinhole 尺寸較小的圖十五，雖然整體光量較低，但沒有明顯的中央亮圓。圖十六是將針孔、毛玻璃配置都調整到最佳化時，所拍攝之影像。

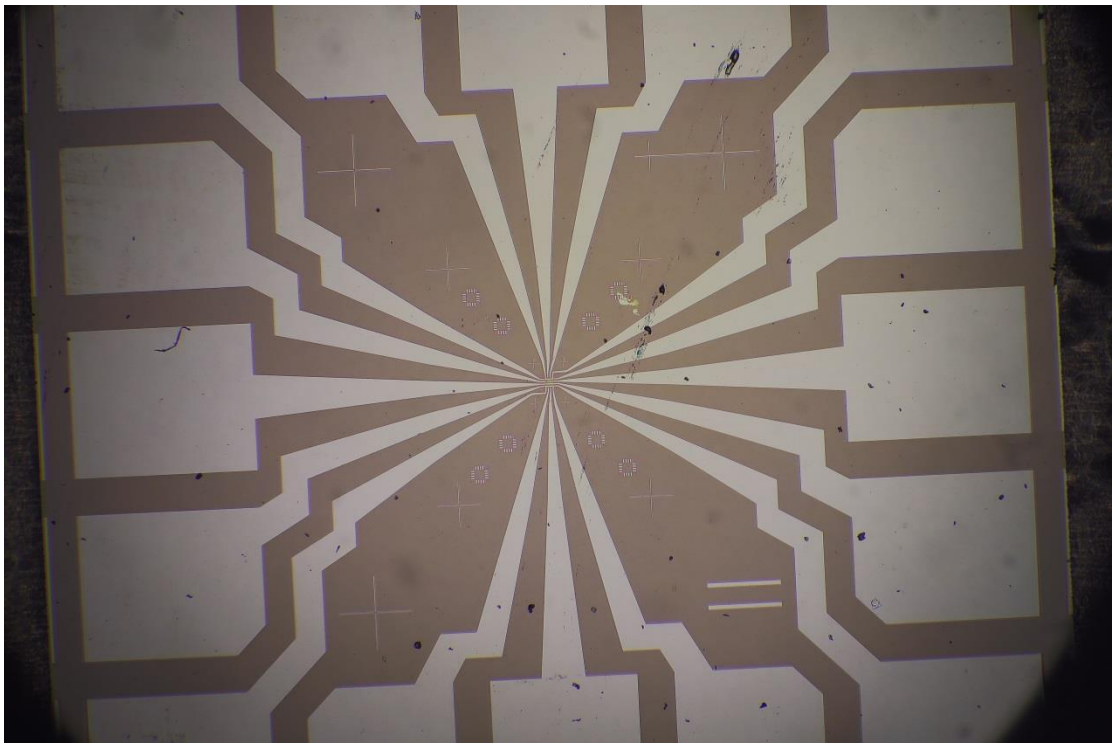


Figure 14 : 200um pinhole

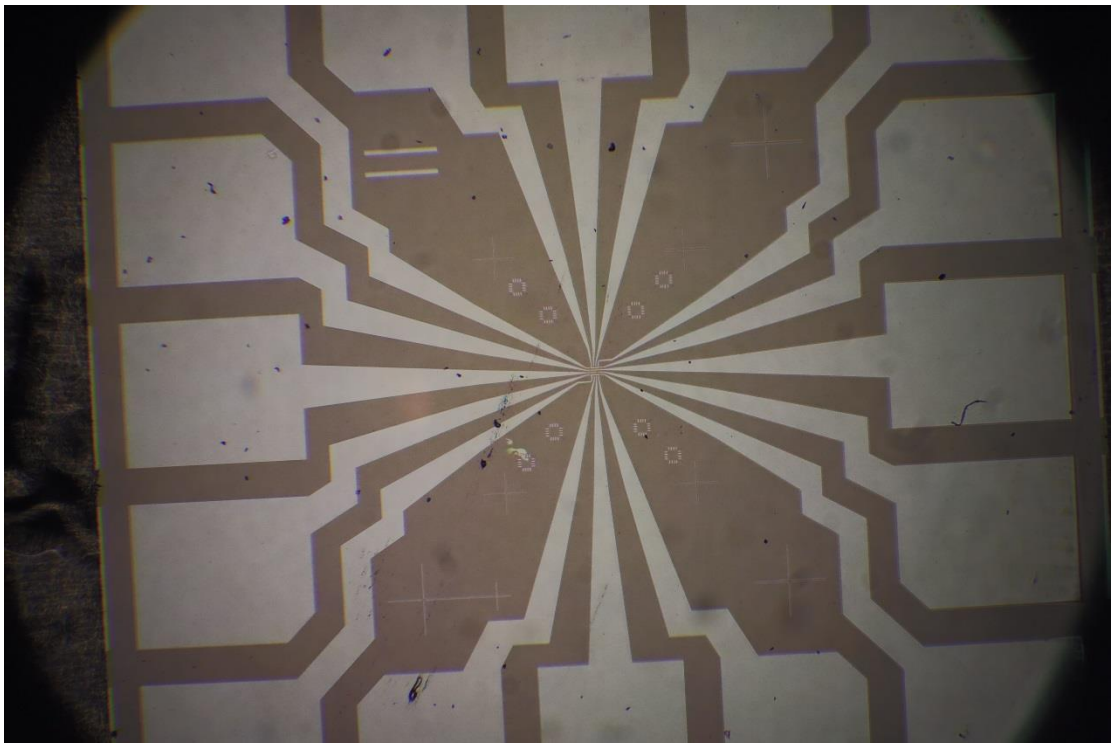


Figure 15 : 100um pinhole

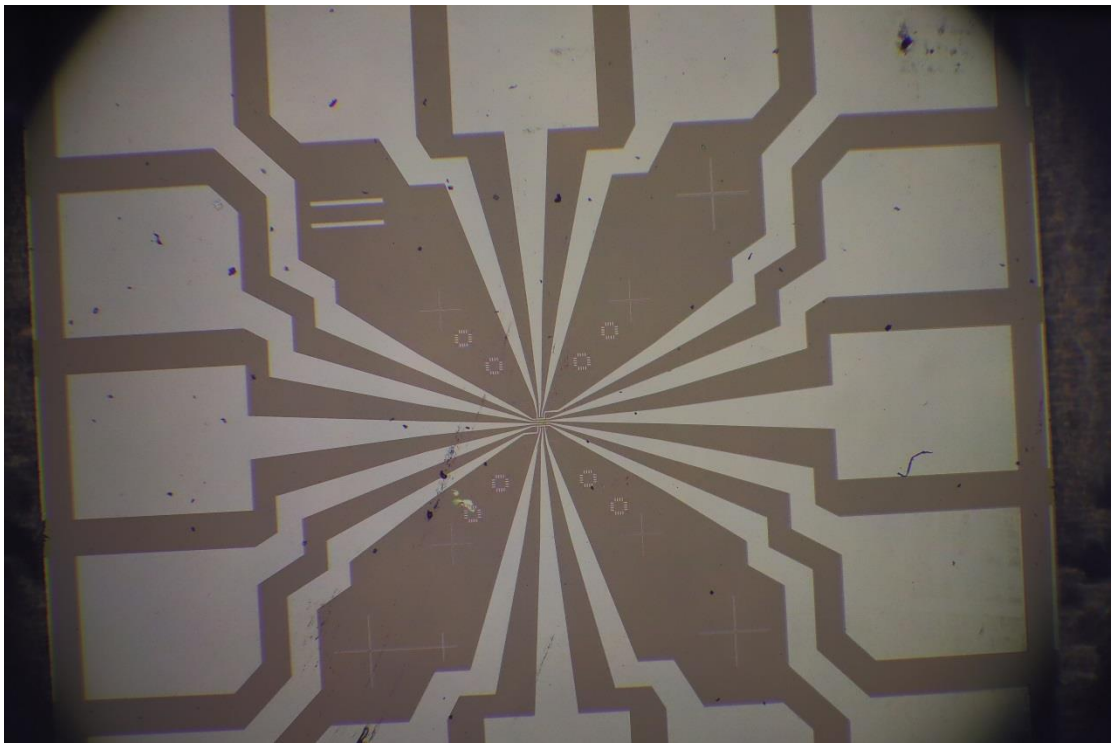


Figure 16 : 100um pinhole, diffuser outside

Aperture

我們希望能在現有的光源配置中在提升照光品質，因此在照明光路中加上一可調光圈。從以下幾張圖片可以看到，雖然調小光圈似乎有助於提升光線的均勻度，但小光圈會直接限縮照光範圍，在目前的一寸管徑照明系統中，沒有辦法達到實質的進步。

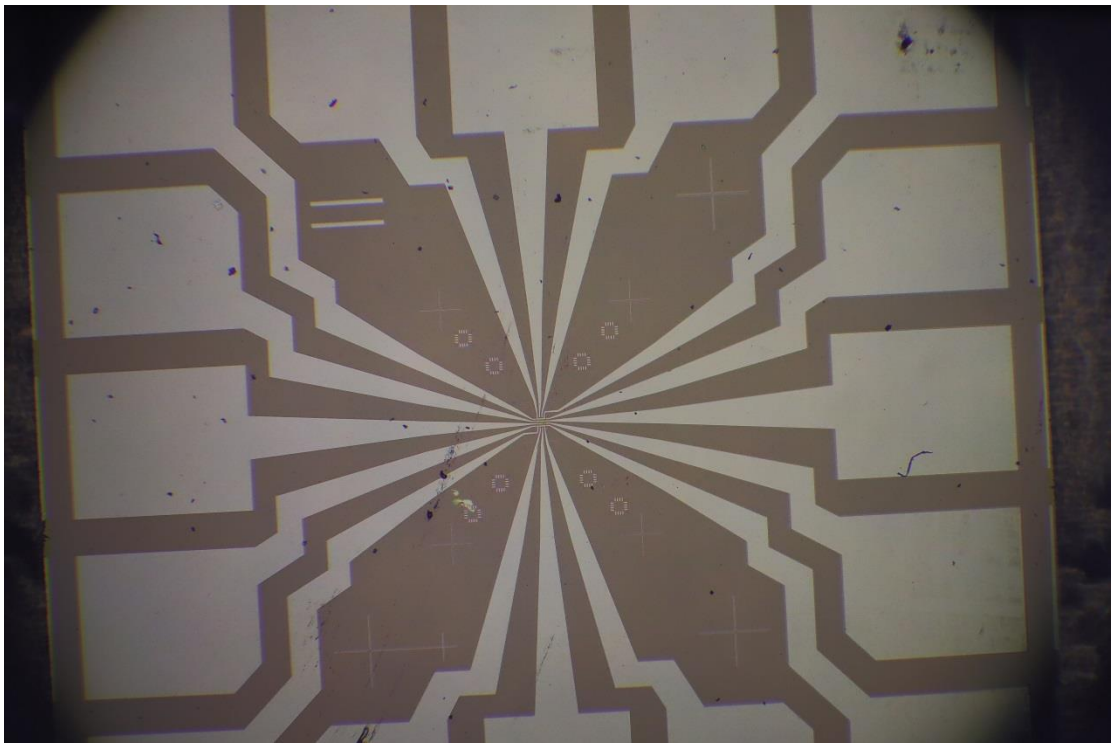


Figure 17： 未加裝光圈

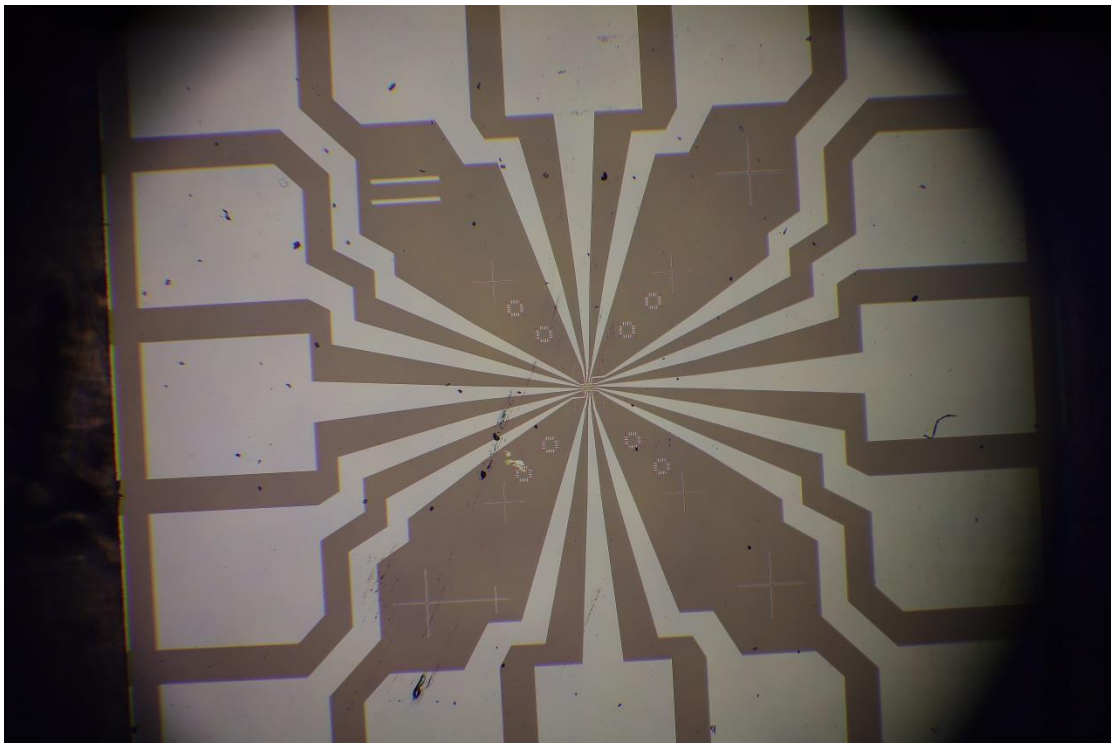


Figure 18： 加入光圈-光圈全開

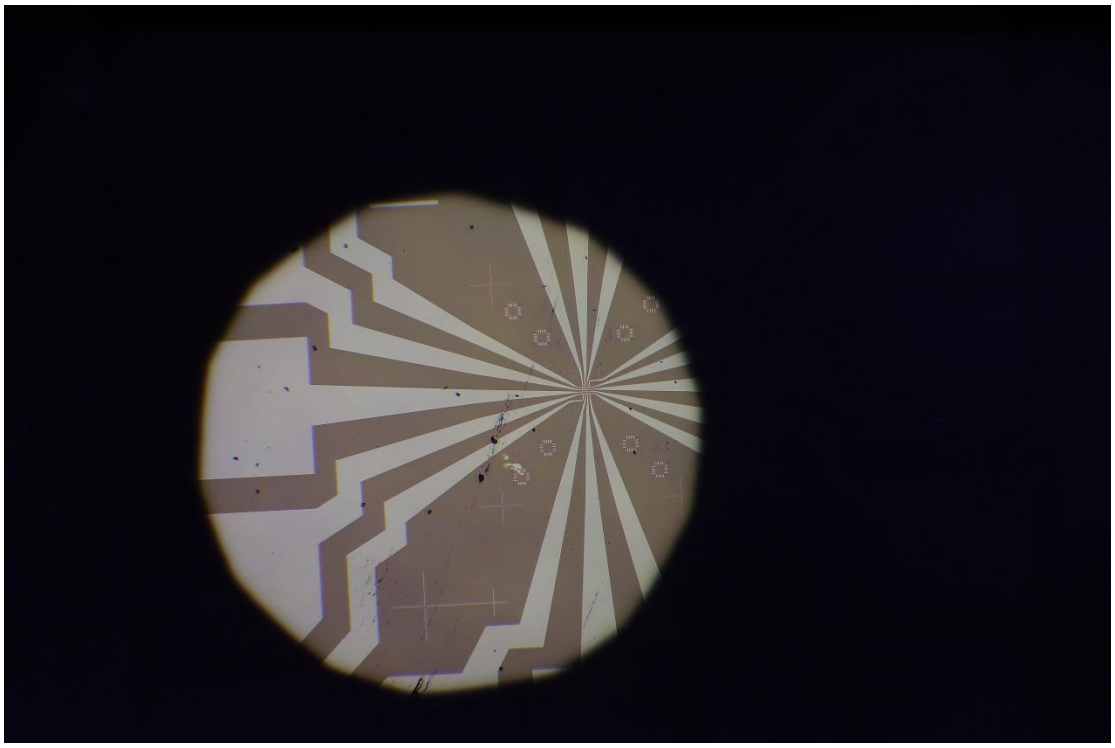


Figure 19： 加入光圈-半開光圈

Figure 20： 加入光圈-最小光圈

SM2

在經過以上嘗試後，我們將整個系統放大，採用兩吋的 SM2 系統，光源部分與原初使用相同的 pinhole 及毛玻璃配置，但將兩面透鏡、直角反射鏡等等都換為兩吋元件。在 SM2 系統中，成像透鏡我們改用 18cm 的 tube lens。

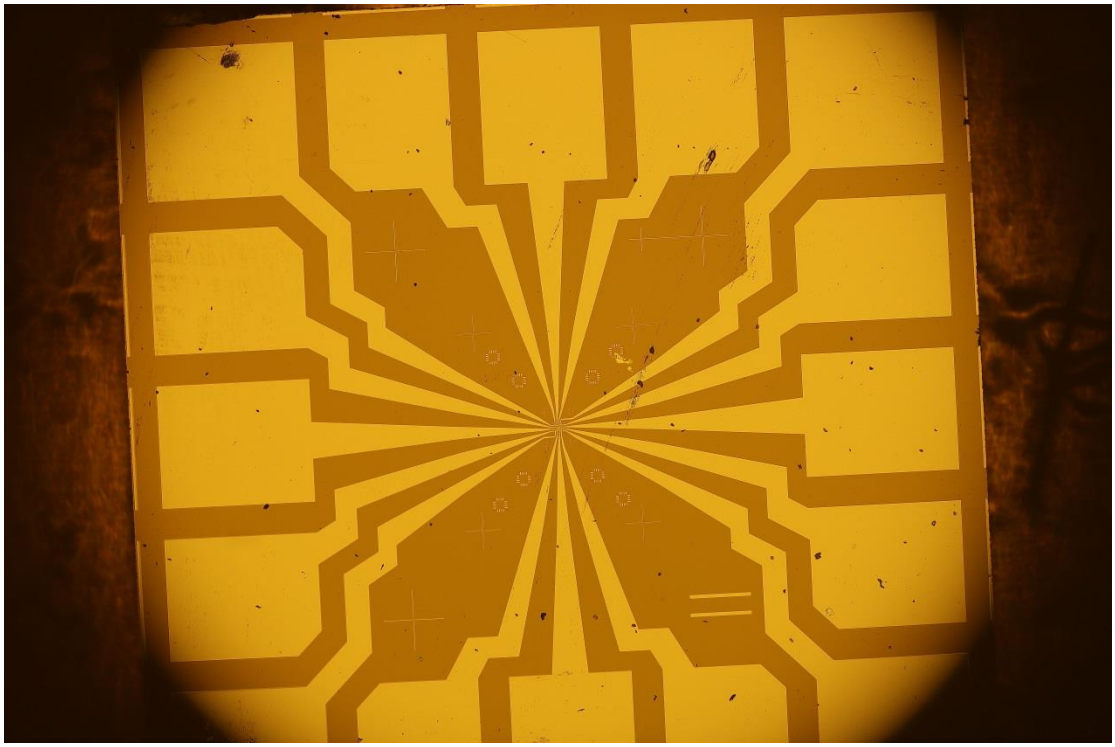


Figure 21 :sm2-no pinhole iso100 1/400

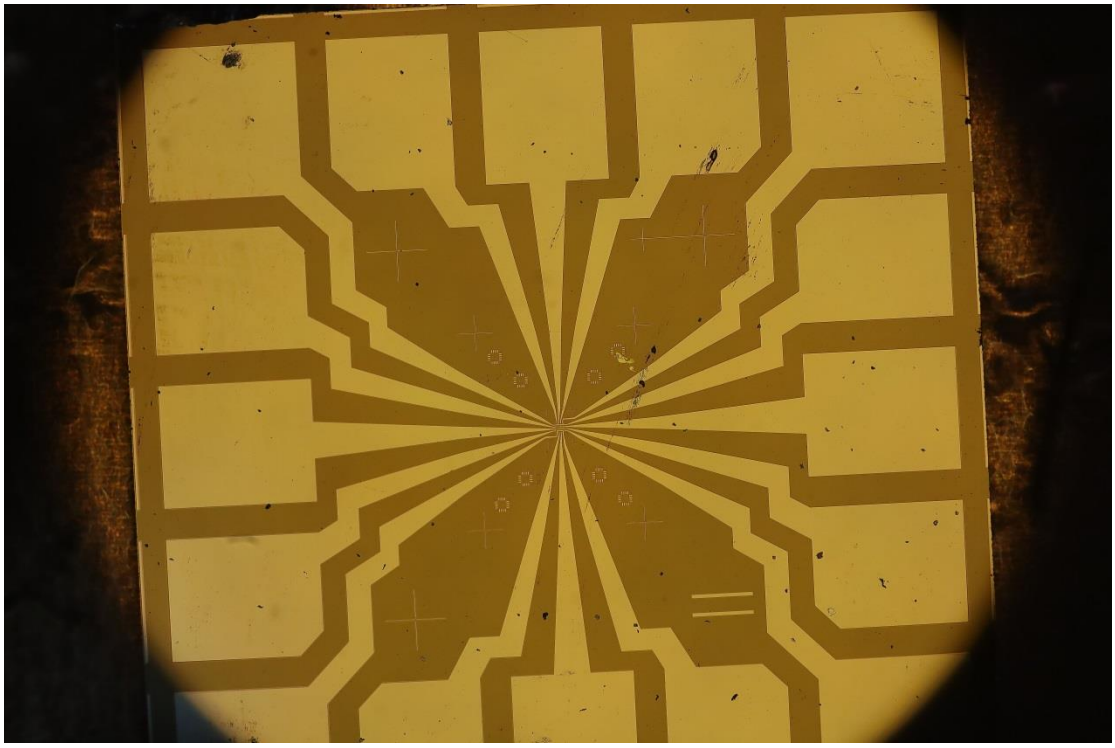


Figure 22 : sm2- 100um pinhole iso3200 1/25

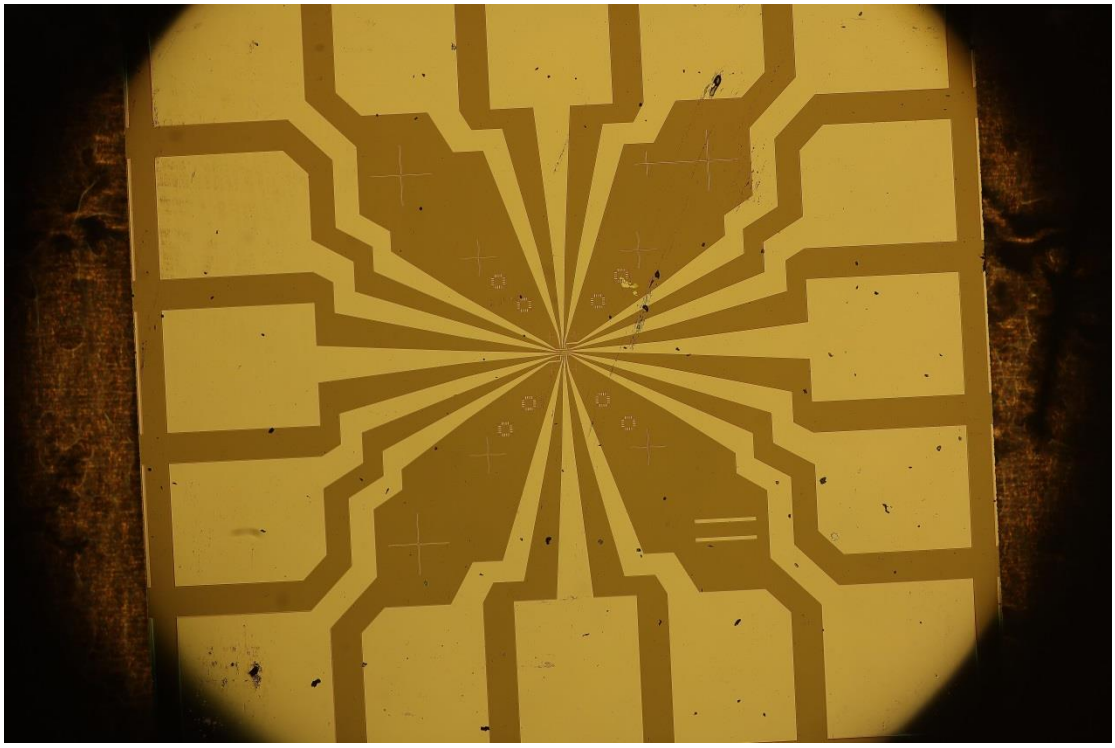


Figure 23 : sm2-200um pinhole iso500 1/15

以上照片都採用四倍物鏡拍攝，但可以看見照光範圍與使用一寸元件時沒有明顯差異。在針孔的使用上，我們發現 100um 與 200um 的針孔都能提升影像的對比度，但 100um 的針孔會使光通量大幅減少，需要較高的 iso 才能得到影像，反而

造成太高的雜訊，因此最後採用 200um 針孔。

SM2 VS. SM1

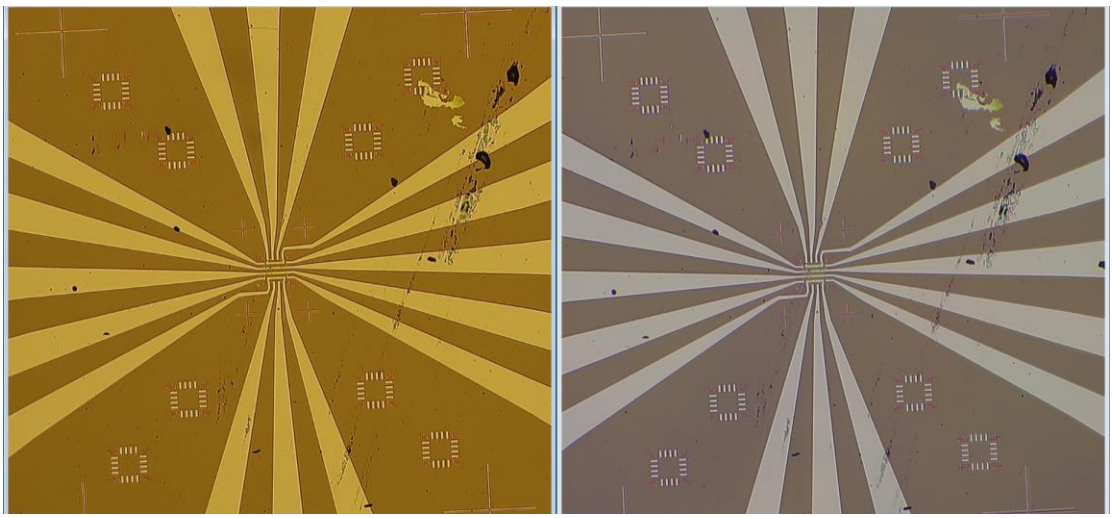


Figure 24 : left-sm2 / right-sm1

從這張比較圖能夠看出，採用 sm2 系統的影像對比度較高，視覺上銳利度也因此較好，而實際所保留的影像細節大致持平，但 sm1 系統的色散較明顯，推測也因為 sm2 系統採用了品質較好的鏡片，且光路的密封程度較好。整體來看，sm2 系統的可是範圍雖然沒有擴大，但影像品質確實較好。

結論與討論

感光元件部分，將原本的 Mightex 1/3” CCD 換為 Canon EOS RP，利用配備有高畫素全片幅 CMOS 影像感測器的相機（全幅機），取得比以往使用小面積 CCD 更廣的視野範圍，同時具有更高的影像解析度。在光路部分，為了最大程度地利用全幅機，將光學系統中的成像透鏡與半反射鏡改為直徑兩吋，讓物鏡視野在全幅機上成像時達到最佳狀態。

為了最大化照光可視範圍，光源部分也做出相應的調整，擴張打光範圍達到全幅機的最大效用。理論上，照光範圍取決於射向物鏡背焦平面的光源入射角（NA），而 NA 則受到光源模組中針孔、透鏡組與管徑的影響。另一方面，照光的均勻度則取決於針孔、光圈與毛玻璃。依據實驗結果，100μm 的針孔照光範圍與均勻度比 200μm 的針孔好。毛玻璃則負責模糊 LED 光源的像，在沒有毛玻璃的狀況下會在影像中看到很清晰明顯的 LED 燈泡成像。而光圈理論上會過濾掉雜光提升

照光均勻度，但是在實驗中卻因為限制了照光範圍，沒辦法提升影像品質。光圈部分推測是因為光圈的放置位置不在光線的焦點上，因此遮擋住了光線。

值得後續討論的議題是，我們將照光系統的管徑全部放大至兩吋，改用 SM2 系統後，照光範圍卻沒有顯著的提升。可能性之一是因為我們在一寸系統就已經達到物鏡的最大照光 NA。

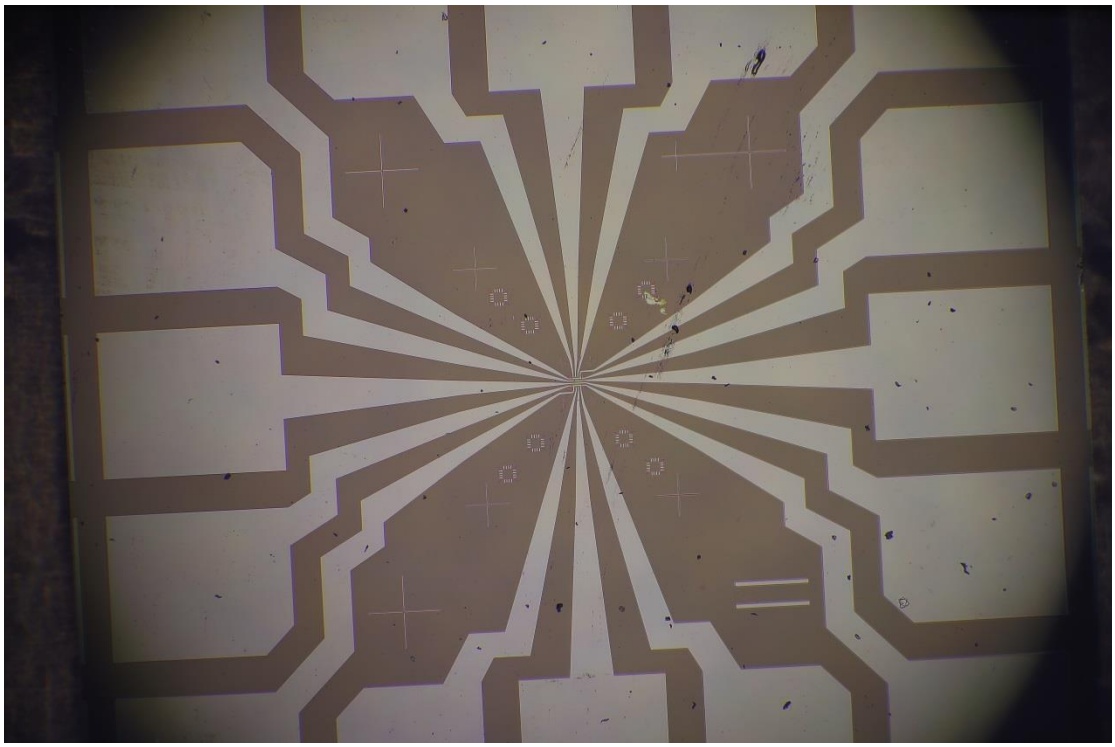


Figure 25 : Final configuration-4x

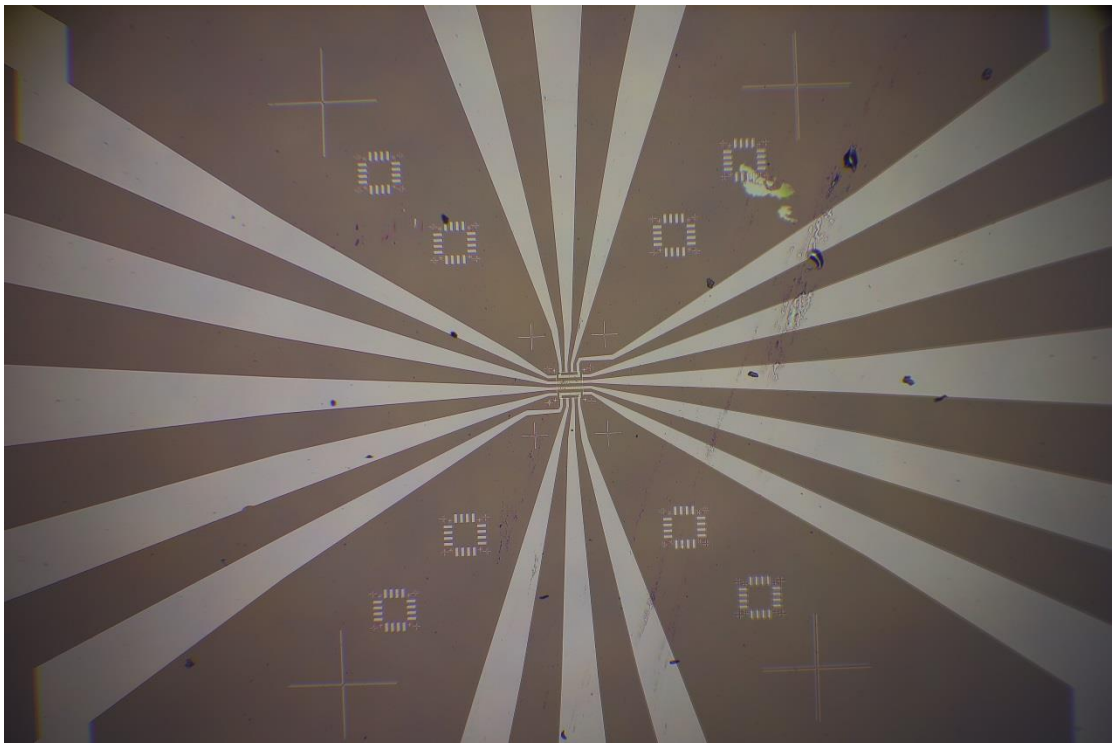


Figure 26 : Final configuration-10x

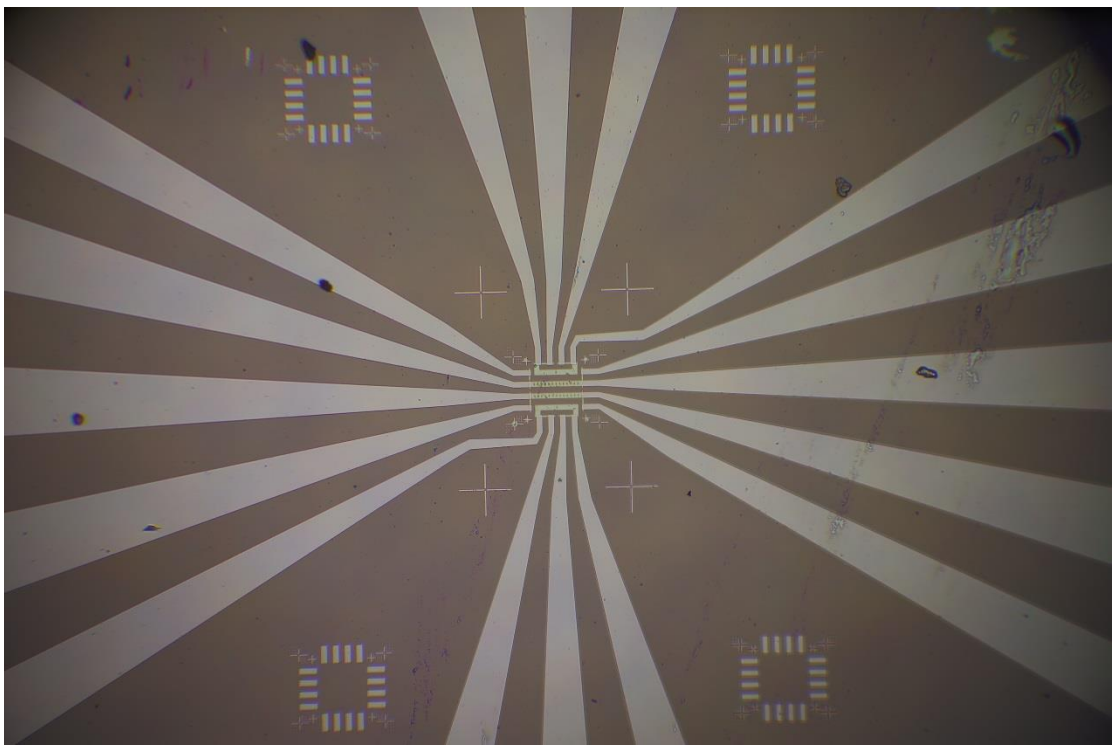


Figure 27 : Final configuration-20x

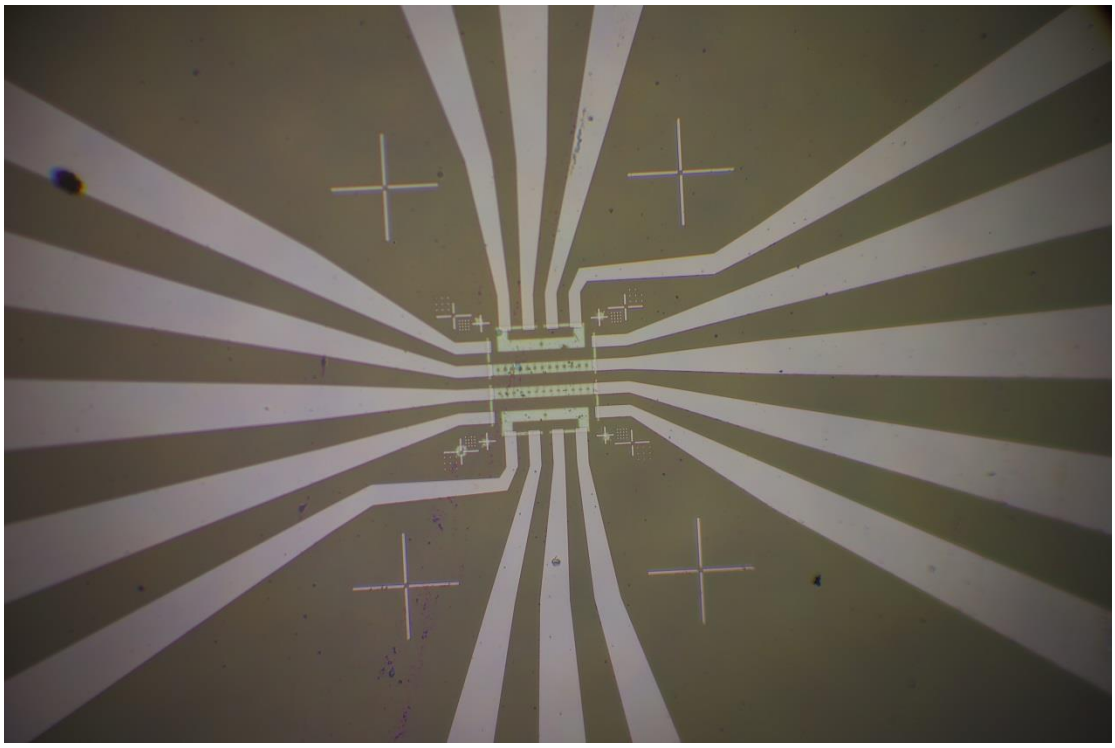


Figure 28 : Final configuration-40x

二、 LabVIEW 控制相機的程式撰寫

Canon 相機操控程式: server\ALL user\2019 大專生暑期實習計畫\鄭泊聲\Canon，除原始碼外，內有 SDK、OM 視野說明等文件
Mightex 相機操控程式: server\ALL user\2019 大專生暑期實習計畫\鄭泊聲\Mightex

SDK 與動態函式庫(.dll)

不論在 Mightex 或是 Canon 系統中，透過 LabVIEW 撰寫操控程式的基本策略都是使用 Call Library Function node 來呼叫 SDK 中的函式。在兩個系統中，SDK 都已經提供了選擇相機、建立相機與電腦的連結、拍攝相片、設定照相參數、獲取影像資料等等的函式。兩套系統的 SDK 說明書都對每個函式有詳細的解說，對於 SDK 的操作邏輯與流程也都有明確的說明，只要依照說明書的方式操作，大致上就能成功地操控相機。

MoveBlock

在兩套系統中，SDK 所提供的函式都能提供影像的 RGB 資料，但函示所傳回的值是 C++語言中的 pointer，代表的是影像在記憶體中的位置，因此 LabVIEW 收到該 pointer 後，必須到該記憶體位置讀取資料(dereference)，在 LabVIEW 中提供這個功能的函示，稱為 moveblock。Moveblock 是 call library function node 中內建的一個函式，共有三個函式參數(argument)，第一個參數接收 C++ pointer，第二個參數是 dereference 的目標資料型態，在兩個系統中，我們都是使用 unsigned 8-bit integer array，第三個參數則是從 pointer 位置開始要進行 dereference 的資料大小，以 byte 為單位。

在 moveblock 的參數設定中，每一個參數都必須精確地符合傳回的 pointer 型態、目標資料的形態及大小，以及呼叫引數的方式，任何一個設定上的錯誤都會造成 LabVIEW 記憶體控管的例外事件(exception)，因而產生 error 1097。Error 1097 是我在程式撰寫中最常遇到的 error code，它代表 labview 所使用的記憶體區塊出現不正常資料取用，一但出現就必須將 LabVIEW 關閉再重新開啟，有時 LabVIEW 甚至會在 error 出現時直接 crush

在 mightex 系統中，moveblock 成功從 pointer 獲取影像 RGB 資料後，我們將資料建立為陣列，此時 RGB 的值分別會占用陣列的一個元素。接著再以 join byte node 將分開的 RGB 值組合為陣列的一個元素，成為能夠繪製成影像的 pixmap，再透過 labview 內建的 unflatten、draw picture 等 node，來建立能夠顯示、儲存的影像資料。

自製.dll

然而，在 Canon 系統中，以 call library function node 叫用 SDK 函式所回傳的 RGB 資料 pointer，無法透過 moveblock 獲取影像資料，總是得到一個所有元素皆為 0 的陣列。然而，一樣的 SDK 函式以 C++語言撰寫所建置的.exe 執行檔，卻能夠正常的獲取 RGB 資料。因此最後我決定以 C++語言撰寫擷取影像所需的程式，將其建置為動態函式庫(Dynamic-Linked Library, dll)，並在 labview 中以 call library function 叫用。如此才能在 labview 中獲得有效的 pointer，並成功獲取影像資料，後續的 join byte 等影像運算，則與 mightex 部分如出一轍。關於為何 labview 無法以 call library function node 叫用 SDK 函式來獲取正確的 pointer，我們推測是因為 call library function node 的參數設定中，無法完整支援 SDK 函式所要求的 pointer of pointer 參數型態。

透過 C++語言所建置的.dll 取代部分 call library function node 來叫用 SDK 函式，還有另一個優點，就是 struct 型態的引數呼叫能夠直接契合 SDK 中的設定。由於 Canon SDK 中有不少函式的參數是 struct 型態，而這樣的參數在 labview 的 call library function node 中，就算搭配 cluster 使用，仍然相當不好設定，很容易出現 invalid parameter 或是 exception 等錯誤。我們推測這是由於 labview 不支援以 reference 呼叫 struct 型態的引數。由於在 C++語言中，能夠直接將 struct 型態的參數以 value 或是 reference 呼叫，因此透過 C++語言撰寫也就能順帶解決這部分的問題。

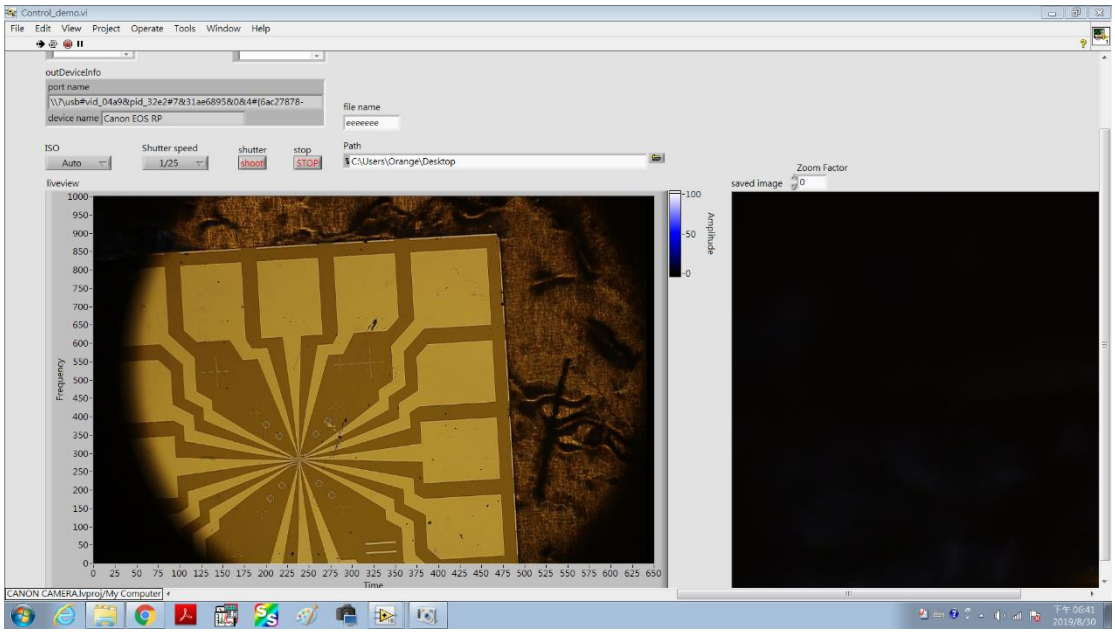


Figure 29： 以 labview 撰寫的 canon 相機操控程式(UI)，能夠設定 ISO、快門、拍照、存檔、liveview

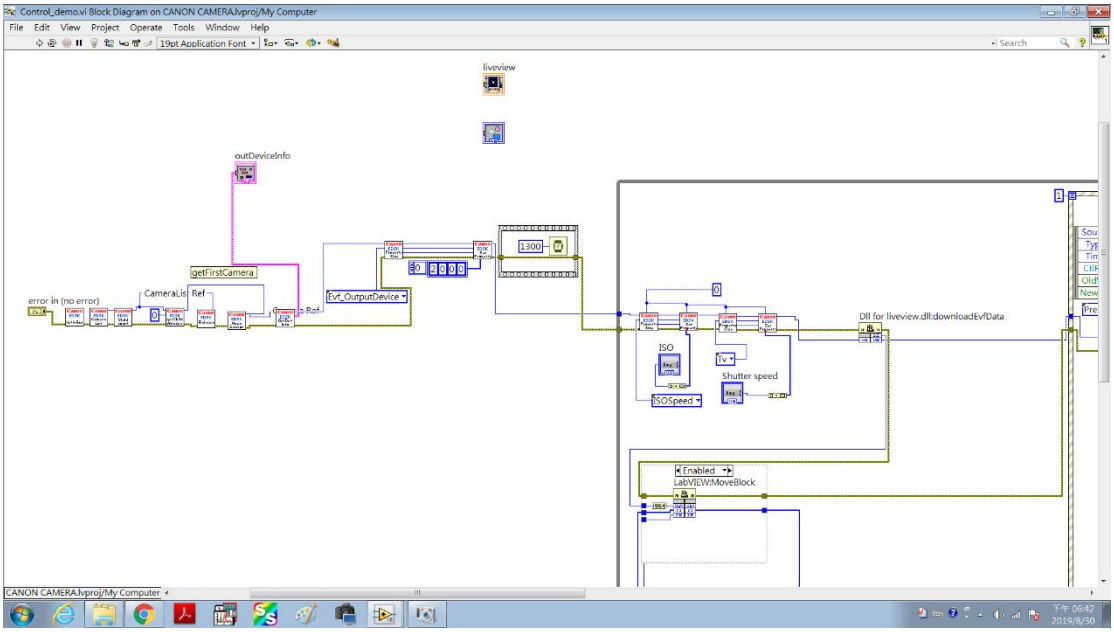


Figure 30： 以 labview 撰寫的 canon 相機操控程式(block diagram 1)

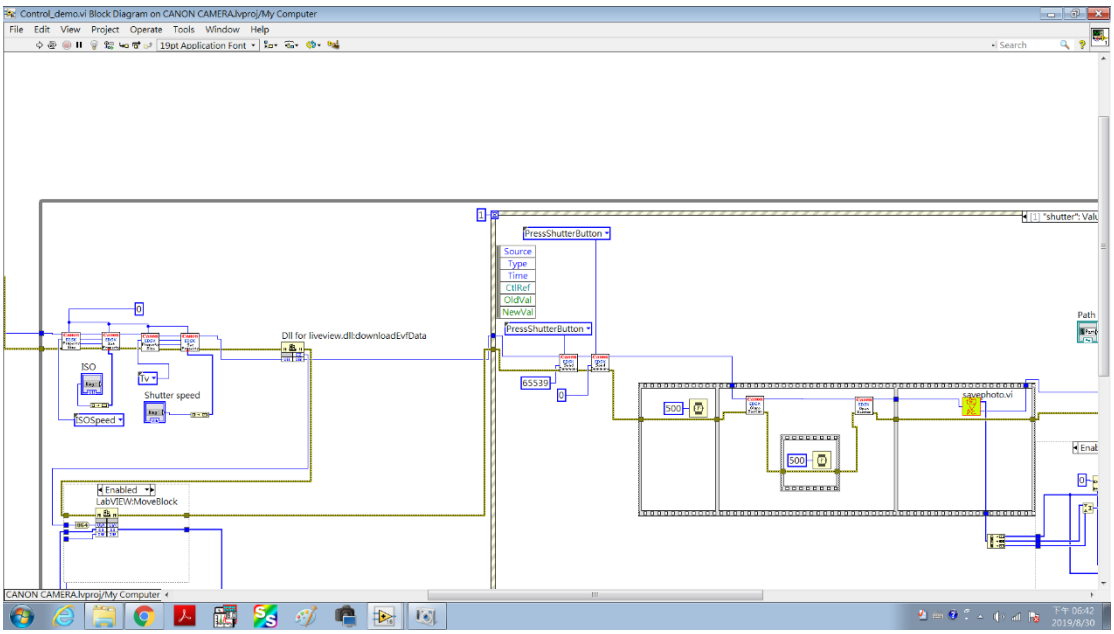


Figure 31： 以 labview 撰寫的 canon 相機操控程式(block diagram 2)

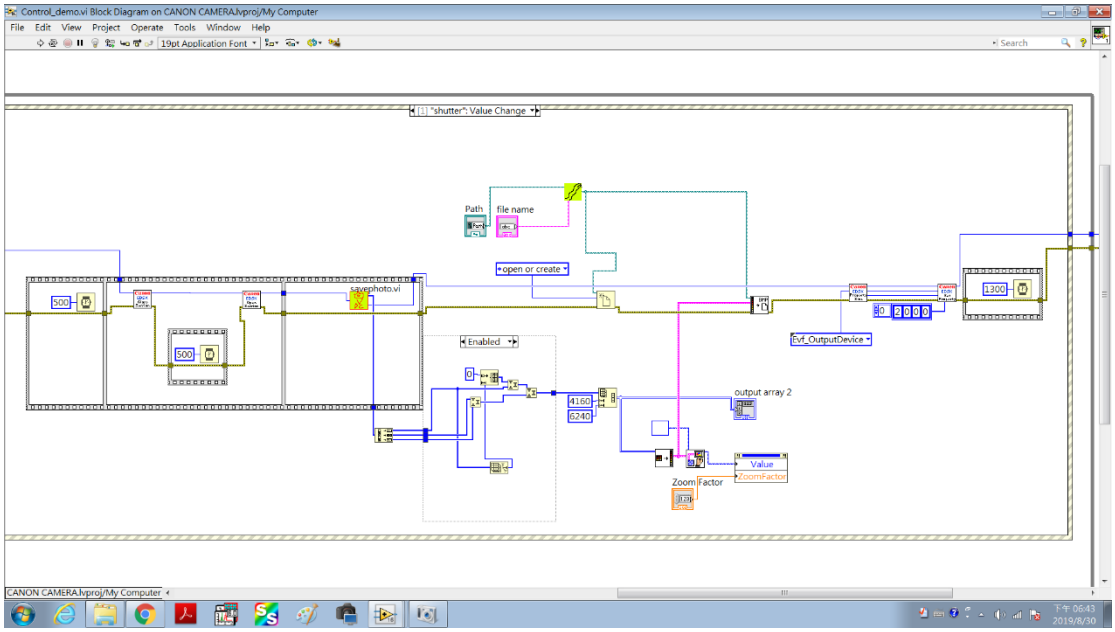


Figure 32： 以 labview 撰寫的 canon 相機操控程式(block diagram 3)

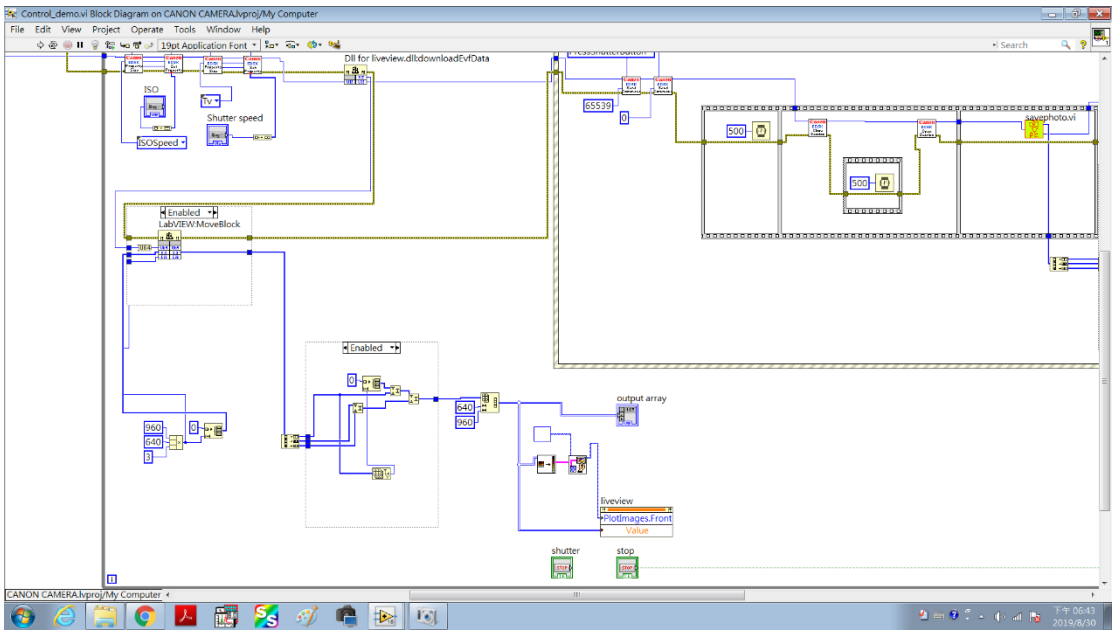


Figure 33： 以 labview 撰寫的 canon 相機操控程式(block diagram 4)

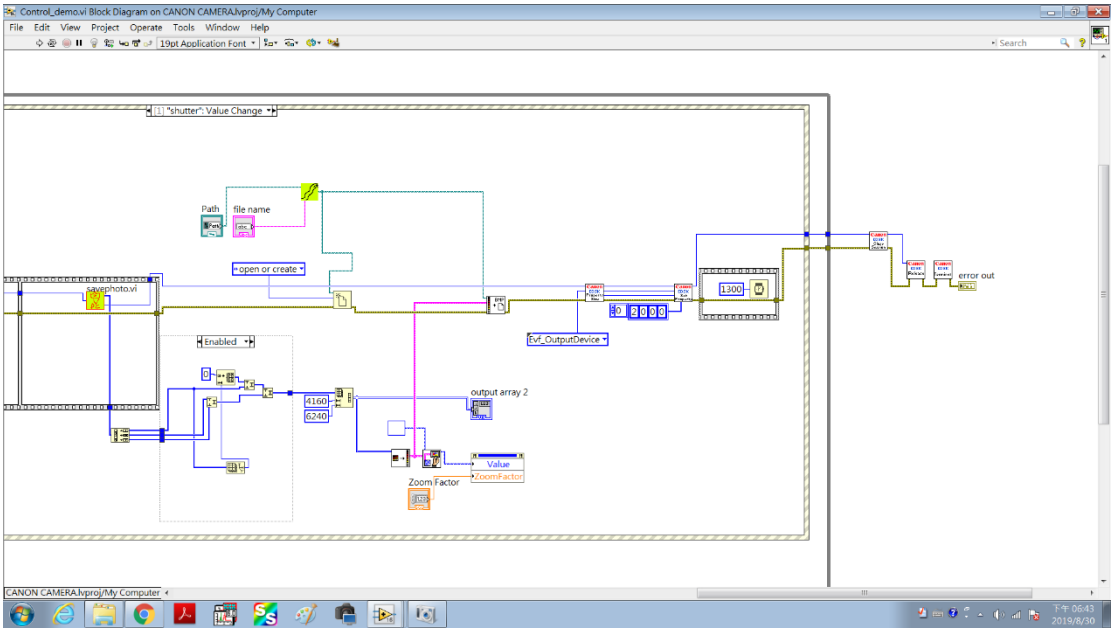


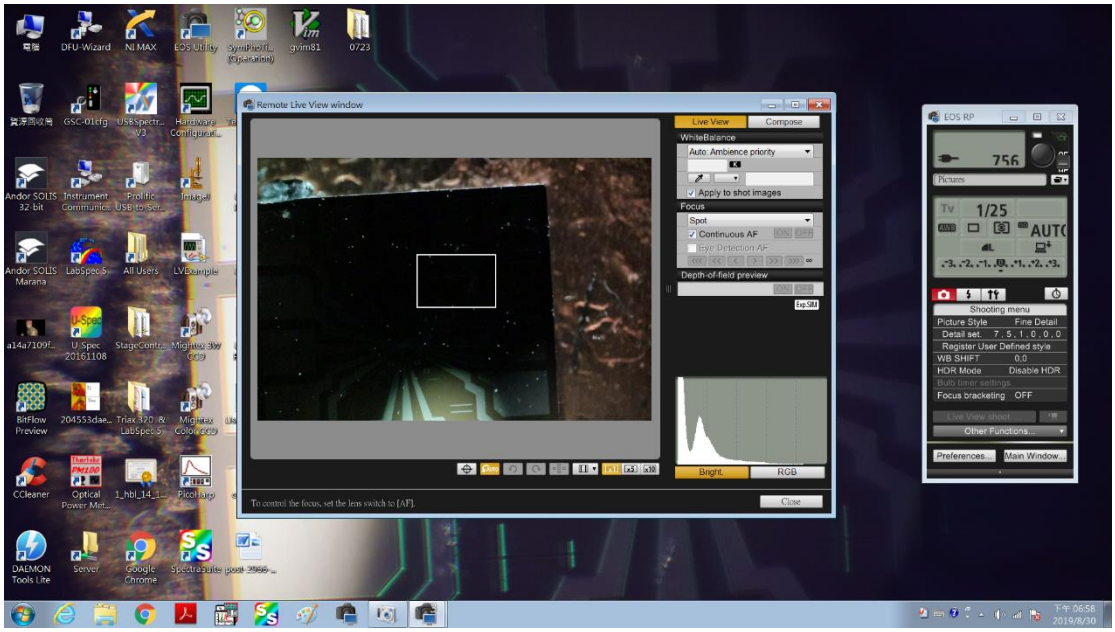
Figure 34： 以 labview 撰寫的 canon 相機操控程式(block diagram 5)

三、 EOS Utility 簡要操作說明

1. 確認相機已經透過 USB TYPE-C 連接口連接至電腦，開啟相機，成功連接後相機螢幕會出現電腦圖示。

2. 成功連接後，開啟 EOS Utility，初始化完能後，點選 “Remote shooting”，就會開啟下圖視窗。

右上角的圓形(數字 754 旁)是快門按鈕，點按即可拍照。按下視窗下半部的 “Live View shoot”按鈕，就會開啟即時影像視窗，如下圖。



3. 關於本 EOS Utility 軟體的安裝，請見 server\ALL user\2019 大專生暑期實習計畫\鄭泊聲\ canon 官方影像擷取軟體的說明。

四、實習心得

本次兩個月的實習經驗，最讓我有收穫的幾個部分，首先當然是熟悉各實驗室常用的 labview 程式開發環境，在程式的開發過程中，我其實也對 C++語言的函式庫、資料結構、資料位置等概念更加熟悉，尤其學習使用 SDK 的開發方式，是相當實用的技能。除此之外，我對影像資料的結構也有更清晰的概念。在光學方面，第一次了解到共阨焦的顯微鏡的原理，並且對我過去在攝影領域對散景、景深、光圈等概念的基礎原理有更深入的了解，格外讓我感到開心。當然，第一次有機會接觸拉曼光譜、雷射等技術，也是寶貴的經驗。

整體來說，這次實習經驗兼具學習與專題開發，幫助我對過去熟悉的領域有更深入的接觸，同時也探索了過去未知的新知識，是一次對我的生涯有全方位助益的經驗。過程中當然要謝謝張玉明老師願意放手給我們空間、讓我們嘗試，也要謝謝每一位實驗室的夥伴在過程中不吝幫助，解答我的各個問題。這次實習小小的遺憾是未能實際操作到影像處理中 lens distortion correction 的部分，不過我已經對影像資料有更近一步的了解，相信未來若能有機會，我也已具備學習這些技能的基礎。