**ADIP 2023 專題報告**

作者一 楊子駿、作者二 葉淞祐

學號 : 112318076 實驗室 : LAB212 指導教授 : 郭天穎

學號 : 111318194實驗室 : LAB212 指導教授 : 郭天穎

**摘要**

**本期末專題的目標是利用Python語言將靜態星空圖像轉化為動態星軌圖GIF。該過程首先牽涉到對星空圖像前景和後景的分離，以便於後續處理。在確定圖像的主要亮點時，我們不僅提供了自動亮度分析來識別最亮的星星，同時也加入了手動選擇功能，允許使用者根據個人偏好選定任意一顆星星作為旋轉的中心點。**

**在自動模式下，系統會分析星空圖中的亮度，自動選出最亮的星星。而在手動模式下，使用者可以自由選擇任何一顆星星作為旋轉的焦點，從而創造出個性化的星軌圖像。**

**實現星軌效果的過程中，我們利用了圖像處理技術和計算方法，讓其他星星繞著選定的中心星星進行圓形運動。整個專案在Visual Studio 2022環境中開發，最終產出既包括展示星星隨時間移動軌跡的GIF格式，也包括靜態的PNG格式圖像。**

**簡介**

在本專題的初階段，我們著重於將星空圖像的前後景進行有效分離。這一過程首先透過Sobel濾波器對邊緣進行銳化，從而使圖像細節更加突出。緊接著，我們應用頻域高通濾波器來保留高頻信號，這進一步提升了邊緣的清晰度。接下來，利用Canny邊緣偵測技術進行更精細的邊緣凸顯，並通過閉運算消除細小的雜點並閉合開口。最終，應用連通元件演算法來確定前景和背景的分界線，並在分界線以上部分應用白色以產生遮罩提取前景。

進入專題的第二階段，我們將關注於辨識並捕捉夜空中最璀璨的星星。這一階段從將圖像轉換為灰度開始，隨後進行二值化處理以凸顯亮點。利用 OpenCV 的 findContour功能在二值化圖像中檢測輪廓，這些輪廓可能代表星星。對每個輪廓進行遮罩處理，以計算出遮罩區域內的最大亮度值及其位置。然後，根據亮度和面積對這些輪廓進行排序，從而從中選出最亮的星星，並記錄其亮度、位置和面積，為最後處理階段奠定基礎參數。

在本專題的最終階段，我們將利用前兩階段生成的遮罩和參數來製作星軌動畫 GIF。首先，程式根據原始星空圖像的尺寸動態調整處理參數，包括 GIF 幀數、持續時間和合成幀數，以適應不同大小的輸入並優化處理時間。接著，利用第一階段生成的前景遮罩來限制星軌僅在背景區域繪製，確保圖像的真實性和美觀度。每一幀的星星位置是通過計算旋轉運動得出的，我們在每組幀中用 cv2.circle 函數繪製星星，進而創造出動態的星軌效果。所有的幀被逐一累加到一張圖像上，這張圖像最終被保存為 PNG 格式。此外，所有累積的幀也被合成為一個 GIF 動畫，從而展現了星星隨時間移動所形成的軌跡。

**提出方法**

**整體架構:**



**圖一 : 整體架構圖**

**第一階段:**

**Sobel濾波器**

Sobel濾波器是一種常用於圖像處理領域的邊緣檢測工具。它運作方式是通過計算圖像亮度的空間導數來突出顯示邊緣。Sobel濾波器特別擅長強調垂直或水平方向的邊緣，因為它分別應用水平和垂直方向的核對圖像進行卷積。這種濾波器特別適用於獲得圖像的粗略邊緣資訊，所以我們先使用他銳化圖像中物件的邊緣，圖三為圖二濾波後的結果。



**圖二(上) : 星空原圖 圖三(下) : Sobel濾波結果**

**頻域高通濾波器**

頻域高通濾波器用於圖像處理中，主要用於去除低頻信息，同時保留高頻信息。在頻域中，圖像被轉換成其頻率成分，高通濾波器屏蔽低頻成分，只讓高頻部分通過。這種方法適合於強調或檢測圖像中的細節和邊緣，所以我們在Sobel濾波器後使用此技術來增強邊緣的細節，如圖四。



**圖四 : HPF結果圖**

**Canny邊緣偵測**

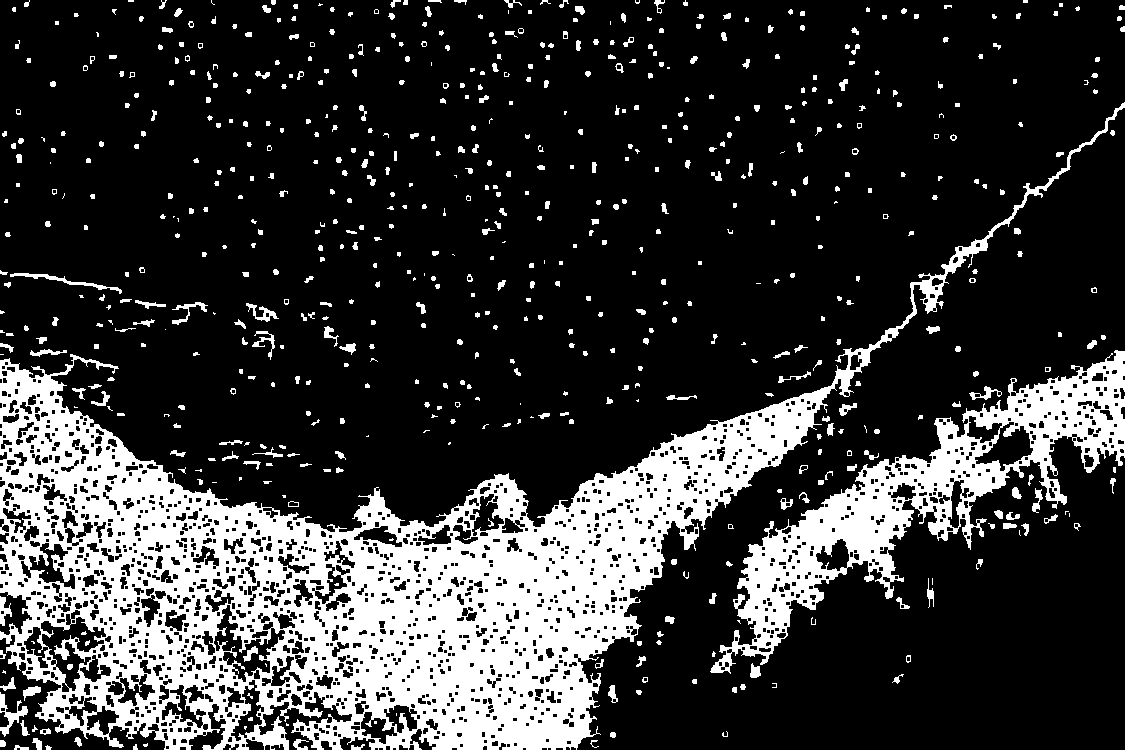
Canny邊緣檢測算法是一種廣泛使用的邊緣檢測技術，被認為是邊緣檢測中的黃金標準。它首先利用高斯濾波器平滑圖像以減少噪聲，然後使用Sobel濾波器計算梯度幅度和方向。隨後，算法應用非最大抑制技術來確定實際的邊緣，並通過雙閾值方法消除弱邊緣，從而實現精確且清晰的邊緣檢測。我們使用他來尋找前後景的交界，如圖五。

****

**圖五 : Canny邊緣檢測結果圖**

**閉運算**

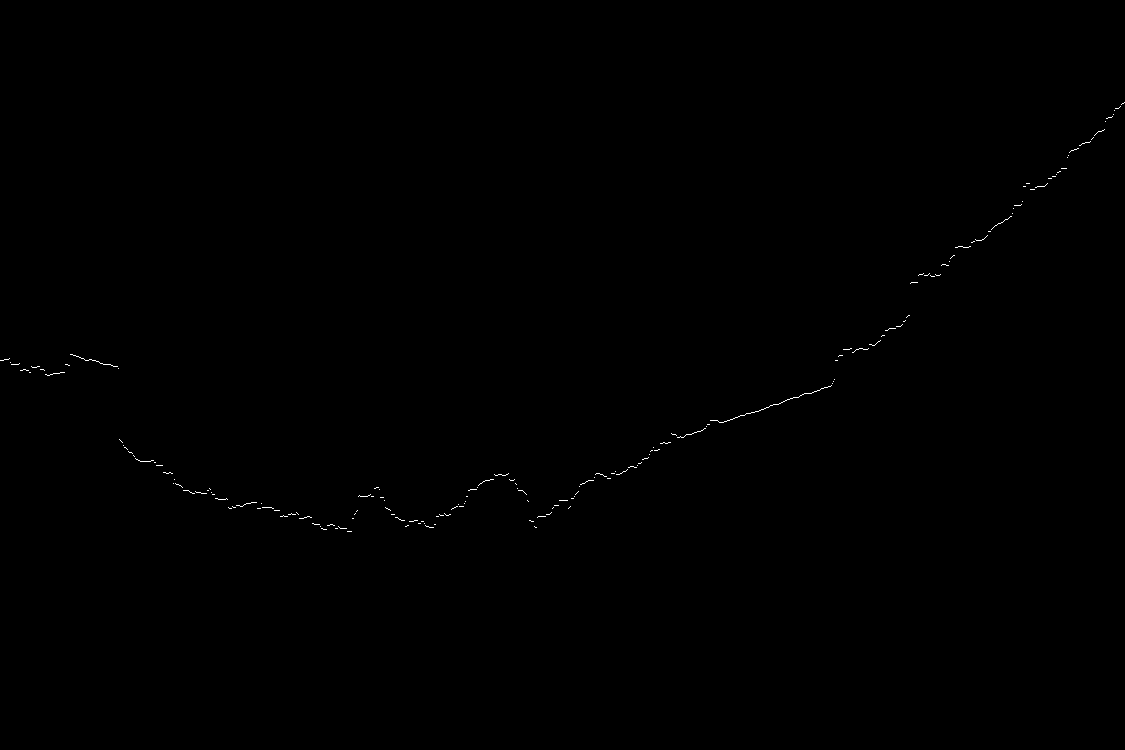
在圖像處理中，閉運算是一種基於形態學的操作，它通常包括先膨脹後腐蝕的過程。這種操作對於平滑物體邊緣、填補小洞和連接靠近的物體特別有效。閉運算可以移除圖像中的小黑點，同時保持圖像主要部分的形狀，因此它在預處理階段中被廣泛用於改善圖像資料。將圖五使用後可以去除不需要的細小點，並將邊緣連接得更順暢，結果如圖六。



**圖五 : Canny邊緣檢測結果圖**

**連通元件分析**

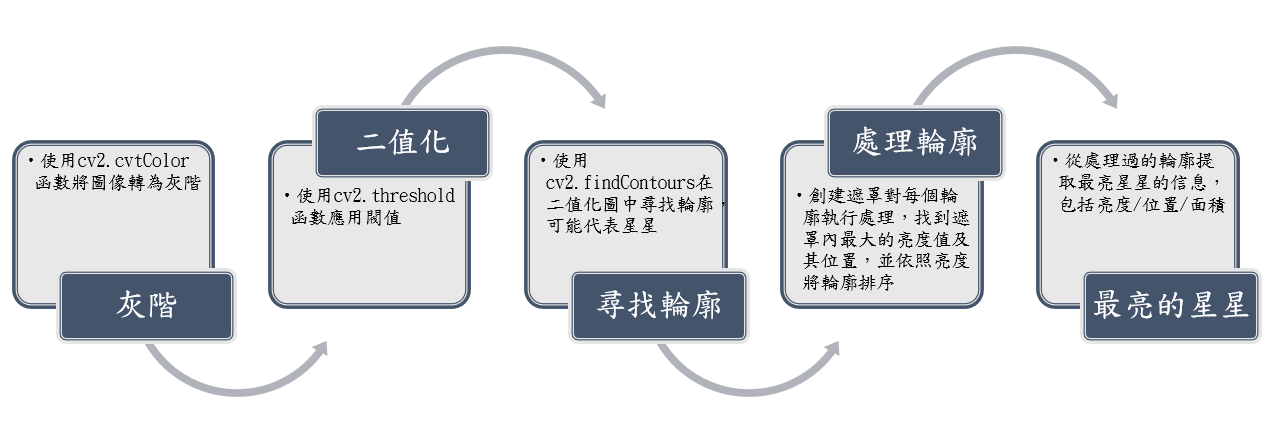
連通元件分析是一種用於分析二值化圖像的技術，它將圖像分割為多個連接的區域。這個過程涉及檢測所有相互連接的像素組並將它們標記為獨立的元件。每個連通元件通常代表圖像中的單個物體或特徵。這項技術在圖像分割、物體識別和計算機視覺中扮演著重要的角色，尤其是在需要從背景中識別和分離出前景物體時，所以我們使用他將前後景的交界獨立標示出來(圖七)以製作前景遮罩，如圖八。



****

**圖七(上) : 連通元件結果圖 圖八(下) : 前景遮罩**

**第二階段:**

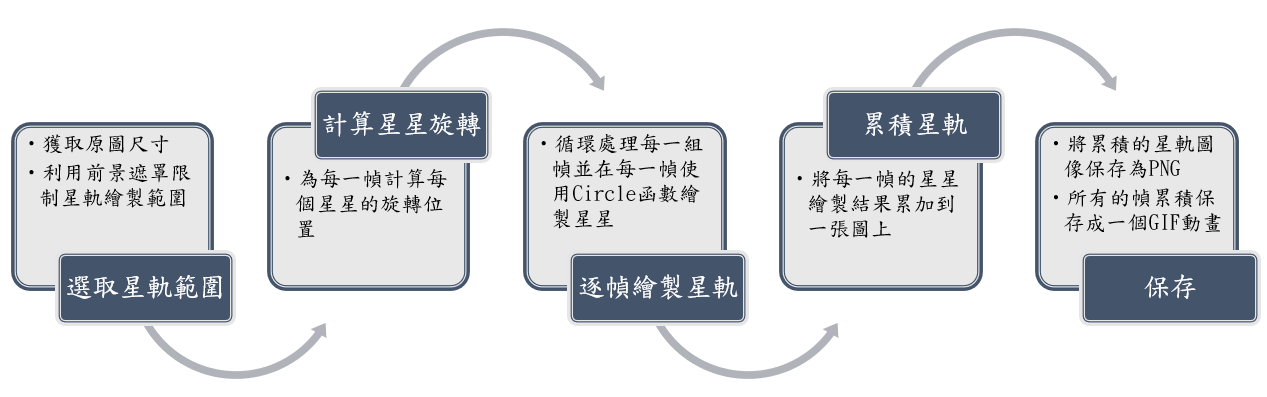
****

**圖九 : 第二階段流程圖**

在專題的第二階段中，我們專注於識別並捕捉夜空中最明亮的星星。這一過程開始於將原始圖像轉換為灰度，然後應用二值化處理以突顯明亮的星星。接著，使用 OpenCV 的 findContours 方法在二值化圖像中檢測可能的星星輪廓。每個輪廓都被賦予遮罩，以進行深入分析，計算遮罩內的最大亮度點及其位置。

進一步的步驟包括對這些潛在的星星輪廓進行排序，根據它們的亮度和面積來確定最亮的星星。這是通過 multiprocessing.Pool 並行處理每個輪廓來提高效率。最終，從這些處理過的輪廓中提取最亮星星的信息，包括其亮度、位置和面積。這些信息對於之後創建星軌圖至關重要，因為它們決定了星軌動畫中星星的初始位置和亮度。

**第三階段:**

****

**圖十 : 第三階段流程圖**

**第三階段:**

在本專題的第三階段，我們專注於利用前兩階段的成果來製作星軌GIF動畫。這一過程的開始是基於原始星空圖像的尺寸來設定關鍵的動畫參數，包括幀數（total\_frames）、每幀的持續時間（duration）以及合成幀數（combine\_frame）。這些參數被細心調整以適應不同大小的圖像，從而確保無論圖像尺寸如何，最終產生的動畫都具有流暢和吸引人的視覺效果。

接下來，利用第一階段創建的前景遮罩，我們確保星軌僅在圖像的背景部分繪製，從而維持視覺上的清晰度和專注度。為了創建動態星軌，程序計算了每個星星隨時間在天空中移動的路徑。這通過對每個星星的位置進行數學旋轉來實現，創造出它們隨時間轉動的錯覺。

隨後，我們在每一幀中用 cv2.circle 函數根據這些計算出的位置繪製星星，並將這些幀逐一累加到兩張圖像上：一張用於累加PNG圖像，另一張用於累加GIF動畫。這種累加過程使得隨著時間的推移，星星在圖像上留下了一條條的軌跡。

最後，將累積的星軌圖像保存為PNG格式，並將所有累積的幀合成為一個迷人的GIF動畫。這一階段的完成不僅展示了技術的精妙，也將觀者帶入了一個動態的夜空之旅，展現了星星隨時間移動形成的絕美軌跡。



**圖十一 : 星軌結果圖**

**實驗結果與比較**

**執行時間:**

在專題中使用三個階段來生成最後的結果，但是第二、三階段涉及大量計算，尤其第三階段生成PNG圖像以及GIF動畫，因此希望減少執行時間。並且為了適應不同大小的圖像都能在短暫時間內生成結果，所以針對不同大小的圖像應用不同設定。

我們採取兩種技術進行加速，第一種針對第三階段進行加速，透過調整動畫生成的總幀數、每幀影像停留時間，以及將多幀合併成一幀來表示，來減少生成GIF動畫的負擔。第二種則是針對第二階段進行加速，使用並行處理使得找尋各星星的輪廓可以利用mutiprocessing套件創建多個進程，每個進程可以同時處理，但此技術沒有應用在第三階段，因為GIF動畫的各幀之間都依賴於前一幀，如果並行處裡會出現錯誤。

表**I**為實驗結果，其中V1表示針對第三階段進行加速的技術，V2表示針對第二階段進行加速的技術，並且應用圖十二進行比較。結果顯示在沒有使用V1、V2時在第二、三階段花費時間很多，而使用V1則大幅減少第三階段時間，最後加上V2使得第二階段減少兩倍多時間。

**表I : 執行時間優化比較圖**

|  |  |  |  |
| --- | --- | --- | --- |
| **使用技術** | **第一階段** | **第二階段** | **第三階段** |
| None | 6.5秒 | 132.5秒 | 1721秒 |
| V1 | 6.68秒 | 131.37秒 | 14.39秒 |
| V1+V2 | 6.49秒 | 52.34秒 | 13.33秒 |



**圖十二 : light\_foreground\_5**

**前景切割困難:**

雖然我們在大多數圖像中都能有效切割出前景，但還是有些例外無法準確切割。

如圖十三中，如果面對前景和背景的顏色接近，也就是兩者間的梯度變化低，導致無法找出前景，因為我們是透過邊緣偵測來找尋前景，為了避免極光、雲朵邊緣也被偵測，因此將邊緣偵測的閥值設定較高。



**圖十三 :前景和背景太暗**

如圖十四中，如果面對背景中的雲朵顏色與天空差異太大，會將雲朵視為前景。



**圖十四 :雲朵與背景顏色差異大**

**結論**

雖然無法將前景精準的切割，但大多數圖像都能找出大概的輪廓，並且也適用於各種場景，像是有極光、亮前景、暗前景的圖像。而在生成GIF中也考量適用於不同大小的圖像，並且搭配加速的設計，使得都能在短時間內生成結果，達到通用性。

**參考資料**

本專題無參考文獻