



DIP Course

Homework Report week #2

Student: p11922004 任祖頤

Outline

- EDGE DETECTION
- GEOMETRICAL MODIFICATION

Sobel edge detection



Sobel edge detection

- 套用 unsharp masking 後進行 sobel edge detection v.s. 直接進行
 - 因為 unsharp masking 會在邊緣形成一個高對比色，進行 sobel edge 時原本只有1個邊界變成3個邊界，因此偵測出來的 edge map 會變得更為複雜



img1_origin_processing



img1_with_unsharp_masking



img2_origin_processing



img2_with_unsharp_masking

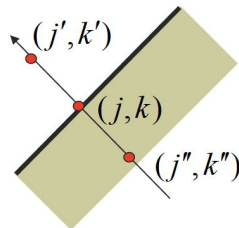


Canny edge detection



Canny edge detection

- 可調控的參數
 - gaussian filter: size = 3, sigma = 1.4 // 影響模糊程度
 - non-maximum supression: dist=2 // (j,k) 到 (j', k') 的延伸距離
 - threshold: candidate 30~149, selected > 150



img_gaussian



img_sobel



img_suppression



img_thresholding

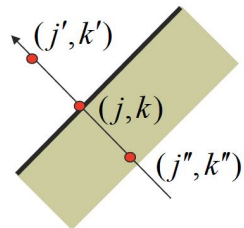


img_result



Canny edge detection

- 參數比較
 - 在 $\text{dist} = 1 \sim 5$ 的調整中, 數字越大, 能被偵測出來 edge 寬度就越大
 - 在 $\text{dist} = 1$ 的情況下因為容許範圍較窄, 很多 edge 因此被截斷
 - $\text{dist} \geq 3$ 的情況下線段較寬, 會看到許多 edge 沾黏在一起
 - $\text{dist} = 2$ 目測是最為完整的



$\text{dist}=1$



$\text{dist}=2$



$\text{dist}=3$



$\text{dist}=4$



$\text{dist}=5$



Canny edge detection

- 參數比較 - 設定 $\text{threshold} = [a, b]$, 其中 a 為 candidate 下限, b 為 candidate 上限, $> b = \text{selected}$
 - 太低的上下限會使得獲選線段過於複雜 (很多重複的線條)
 - 因為演算法是由 selected pixel ($v=255$) 的每個點去出發去尋找 candidate, 因此上限設定太高會使出發點太少, 結果也不佳 (如 $[100, 220]$, $[150, 240]$)
 - 這邊找到比較合適的設定為 $[30, 150]$

threshold=[10, 100]



threshold=[30, 150]



threshold=[50, 200]



threshold=[100, 220]



threshold=[150, 240]

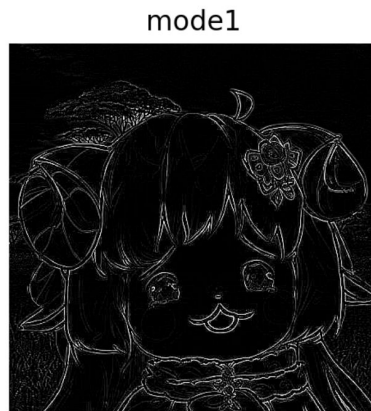


Edge map of Laplacian of Gaussian edge detection

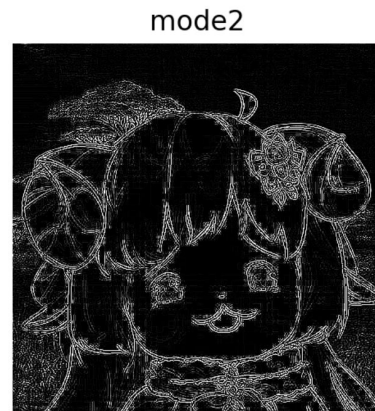


Edge map of Laplacian of Gaussian edge detection

- 可調控的參數
 - **mode1 filter:** 因為 filter 偵測的對比度要求更高, 因此只有比較明顯的黑色線段會被檢查出來
 - **mode2 filter:** 因為 filter 較平滑, 因此許多色差較低的色塊也被判斷為邊界, 雜訊較多



```
[-1, -1, -1],  
[-1, 8, -1],  
[-1, -1, -1]
```



```
[0, -1, 0],  
[-1, 4, -1],  
[0, -1, 0]
```

Edge crispening



Edge crispening



mode1



```
[0, -1, 0],  
[-1, 5, -1],  
[0, -1, 0]
```

mode2



```
[1, -2, 1],  
[-2, 5, -2],  
[1, -2, 1]
```

mode3



```
[1, -3, 1],  
[-3, 9, -3],  
[1, -3, 1]
```

Edge crispening

- 三種 filter 的比較
 - **mode1**: 與 mode3 的結果相似
 - **mode2**: 較 mode1 與 mode3 結果更差, 線條更模糊一點
 - **mode3**: 三種 filter 中邊緣銳利化的效果最好, 因為他更大幅度的提升高低 值之間的落差

mode1



mode2



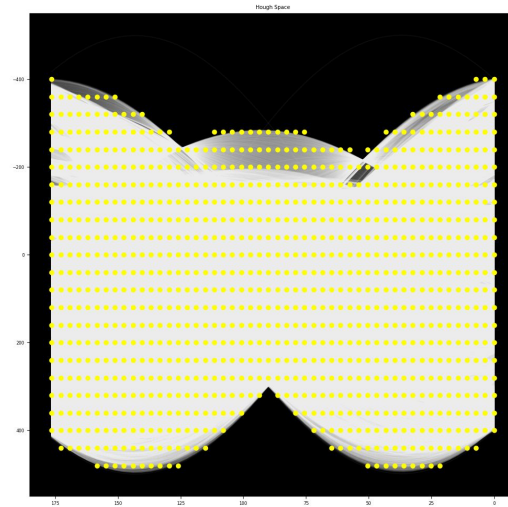
mode3



The Hough transformation of Canny edge detection



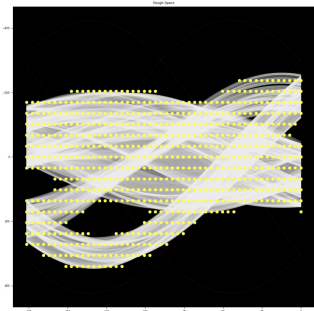
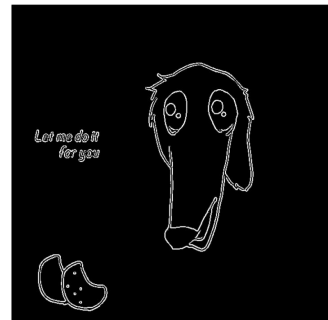
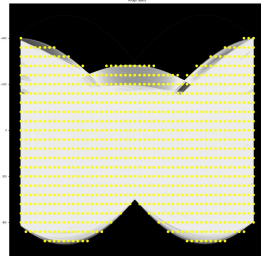
The Hough transformation of Canny edge detection



The Hough transformation of Canny edge detection

- Hough transformation

- 用途: 可以用來檢查特徵 (每個 edge 的兩端點), 在空間中的分佈狀況
- sample2.png 因為是一個對稱的圖片, 因此特徵分佈圖也會呈現對稱狀態, 但因為線段較多且密集, 因此 hough space 顏色非常飽滿
- 若使用簡單一點的圖片, 例如 sample3.png 來轉換 hough transformation 就比較容易看出特徵的分佈狀況

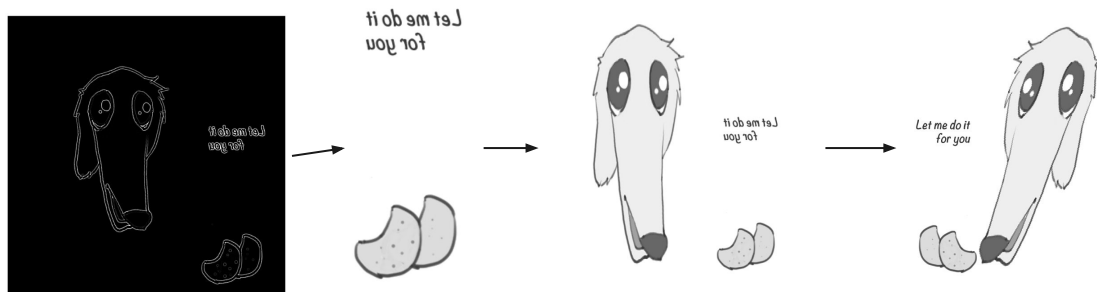


GEOMETRICAL MODIFICATION - I



GEOMETRICAL MODIFICATION - I

- 操作步驟
 1. 先使用 log_filtering 取得原圖的 edge map
 2. 沿著邊緣向內尋找 edge (pixel value=255), 取得文字與洋芋片的保留區域
 3. 使用 scaling 技巧將狗狗拉長(對應到保留區域則不處理)
 4. 使用 affine transformation 將狗狗拉斜(對應到保留區域則不處理)



GEOMETRICAL MODIFICATION - II



GEOMETRICAL MODIFICATION - II

- 操作步驟

1. 先尋找中心點, 這邊取狗嘴的中心
2. 預設半徑為 $r = 80\text{px}$, 每一個迴圈 -1px , 每個迴圈旋轉的密度 $\text{density} = 10000$ (取一萬個點)
3. 預設距離為 $\text{range} = 0\text{px}$, 每一個迴圈 $+1\text{px}$
4. 對每個點取

$$u = \cos(\theta) * (r - \text{range}) + \text{center}(y)$$

$$v = \sin(\theta) * (r - \text{range}) + \text{center}(x)$$



Thank you