視覺感測技術應用實務

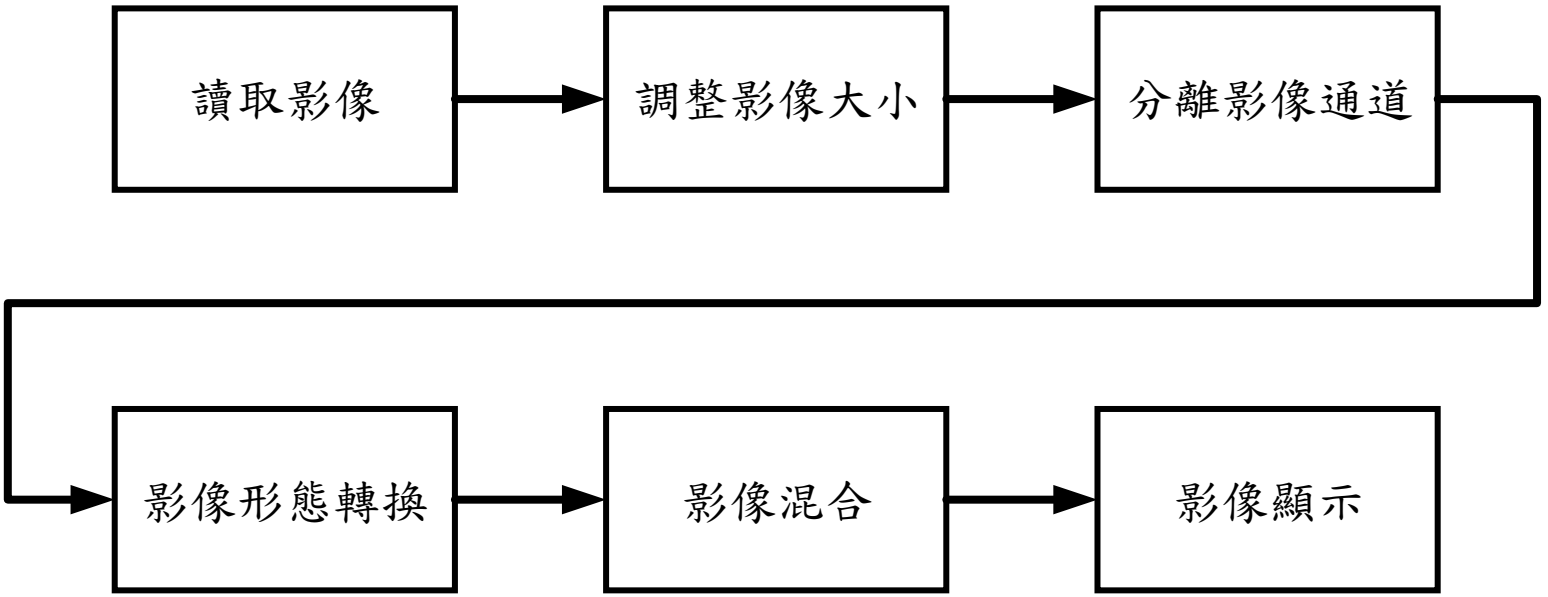
影像混合顯示功能
第二組

組員:李宗晏 608470042(100%)

目錄

- 實現方法
- 獨特設計之處
- 結果展示
- 組員分工表

實現方法--動作流程圖



實現方法--使用類別方法的類型

- 本次程式BaseIP和AlphaBlend的成員函式都使用static method，因為本次類別的成員函式不需要像是self或是cls等實例或類別的參考，所以使用static method可以比較簡明且有效率。
  - 簡明的部分在於不需要多接收一個無關緊要的引數
  - 效率在於一般的實例方法是bound method(是個object)且是在我們要使用他的時候才生成，這會多花費一點的時間，而靜態方法並不會

實現方法--程式(cv2IP.py)

BaseIP類別

cv2IP.py | cv2IPApp.py | Project1.py

cv2IP.py > ...

```python
#!/usr/bin/python3

import cv2
import numpy as np
import enum

class BaseIP(object):
    Obj_Num = 0

    def __init__(self):
        BaseIP.Obj_Num += 1
        print("Create 1 obj: Total number of BaseIP objects is "+ str(BaseIP.Obj_Num))

    def __del__(self):
        BaseIP.Obj_Num -= 1
        print("Delete 1 obj: Total number of BaseIP objects is "+ str(BaseIP.Obj_Num))

    @staticmethod
    def ImRead(filename):
        return cv2.imread(filename, cv2.IMREAD_UNCHANGED)

    @staticmethod
    def ImWrite(filename, img):
        cv2.imwrite(filename, img)

    @staticmethod
    def ImShow(winname, img):
        cv2.imshow(winname, img)

    @staticmethod
    def ImWindow(winname):
        cv2.namedWindow(winname, cv2.WINDOW_NORMAL)
```

AlphaBlend類別的繼承與建構式

```python
class AlphaBlend(BaseIP):

    def __init__(self):
        super().__init__()

```

實現方法--程式(Project1.py)

主函式

cv2IP.py | cv2IPApp.py | Project1.py ×

Project1.py > MyAlPhaBlend

```
1  #!/usr/bin/python3
2
3  import cv2
4  import numpy as np
5  import cv2IP
```

```
60
61  if __name__ == '__main__':
62      MyAlPhaBlend()
63
```

創建AlphaBlend的物件

```
6
7   def MyAlPhaBlend():
8       IP = cv2IP.AlphaBlend()
9
```

- 使用物件導向方式來呼叫類別中的成員函式

實現方法--程式(Project1.py)

- 讀檔程式

```
10      SrcImg = IP.ImRead("img/ghost.png")
11      SrcImg = cv2.resize(SrcImg, (240, 200))
12      back = IP.ImRead("img/background.png")
```

- 前景圖 SrcImg(240, 200)

- 背景圖 back(1050, 1680)

實現方法--程式

- 分離通道程式(Project1.py)

```
13
14      fore, alpha = IP.SplitAlpha(SrcImg)
15      if back.shape[2] == 4: # channels: b, g, r, alpha
16          back = back[:,:,:3]
```

- SplitAlpha成員函式(cv2IP.py)

```
40      @staticmethod
41      def SplitAlpha(SrcImg):
42          fore = cv2.merge([SrcImg[:,:,0], SrcImg[:,:,1], SrcImg[:,:,2]])
43          alpha = cv2.merge([SrcImg[:,:,3], SrcImg[:,:,3], SrcImg[:,:,3]])
44
45          return fore, alpha
46
```

實現方法--程式

- fore(b, g, r)

- alpha(alpha, alpha, alpha)

實現方法--程式

- 轉換影像資料型態(Project1.py)

```
17
18      fore = np.float32(fore) # convertTo 32FC3
19      alpha = np.float32(alpha) / 255.0 # convertTo 32FC3 and normalize
20      back = np.float32(back)
21
```

- fore跟back
  - 資料型態
    - uint8 => float32
  - 數值範圍
    - 0.0 ~ 255.0
- alpha
  - 資料型態
    - uint8 => float32
  - 數值範圍
    - 0.0 ~ 1.0

實現方法--程式(Project1.py)

- 影像混合跟前景在背景中向右移顯示

```
22      k = 0
23      while k != 13:
24          i = 0
25          while i < 4 and k != 13:
26              out = np.array(back)
27
28              rows_start = 770
29              rows_end = 770 + fore.shape[0]
30              columns_start = 350 + 350 * i
31              columns_end = 350 + fore.shape[1] + 350 * i
32
33              out[rows_start:rows_end, columns_start:columns_end]= IP.MyDoBlending(
34                  fore, back[rows_start:rows_end, columns_start:columns_end], alpha, 0.25*(1+i))
35              out = np.uint8(out)
36
37              IP.ImWindow("AlphaBlending Result")
38              IP.ImShow("AlphaBlending Result", out)
39              k = cv2.waitKey(1000)
40              i += 1
41
```

實現方法--程式(Project1.py)

- 影像混合跟前景在背景中正中央顯示

```
42          if k != 13:
43              fore_final = cv2.resize(fore, (1080, 880))
44              alpha_final = cv2.resize(alpha, (1080, 880))
45              out = np.array(back)
46
47              rows_start = int(back.shape[0] / 2 - fore_final.shape[0] / 2)
48              rows_end = int(back.shape[0] / 2 + fore_final.shape[0] / 2)
49              columns_start = int(back.shape[1] / 2 - fore_final.shape[1] / 2)
50              columns_end = int(back.shape[1] / 2 + fore_final.shape[1] / 2)
51
52              out[rows_start:rows_end, columns_start:columns_end] = IP.MyDoBlending(
53                  fore_final, back[rows_start:rows_end, columns_start:columns_end], alpha_final, 1)
54              out = np.uint8(out)
55
56              IP.ImWindow("AlphaBlending Result")
57              IP.ImShow("AlphaBlending Result", out)
58              k = cv2.waitKey(1000)
59
60      del IP
61
```

獨特設計之處(cv2IP.py)

- MyDoBlending跟DoBlending成員函式

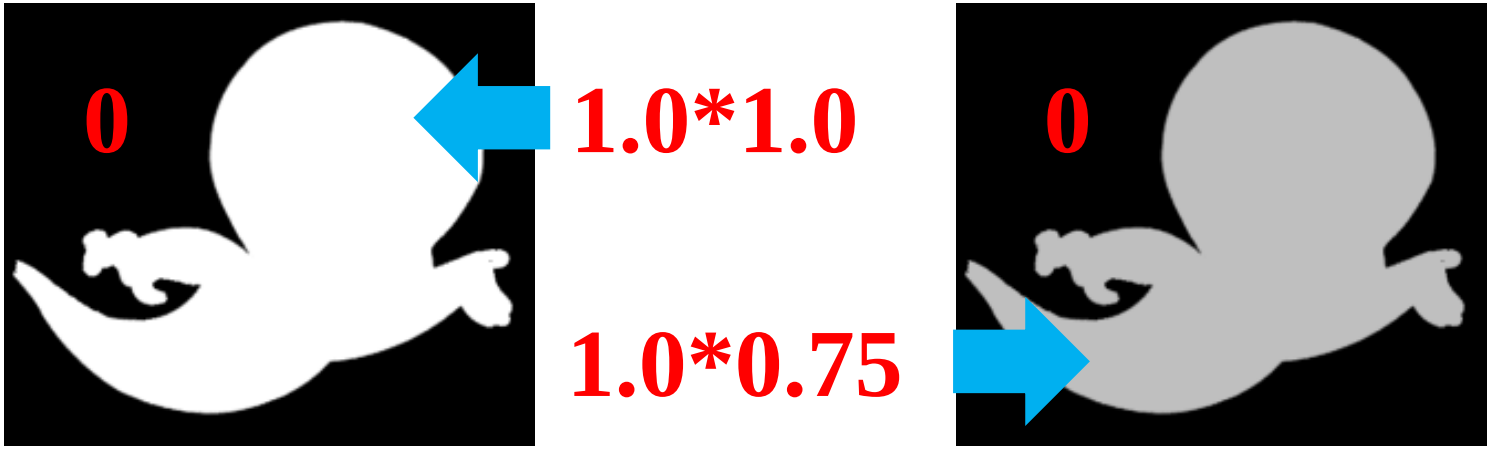
```
47      @staticmethod
48      def DoBlending(Foreground, Background, Alpha):
49          fore = Foreground * Alpha
50          back = Background * (1.0 - Alpha)
51          out = fore + back
52          return out
53
54      @staticmethod
55      def MyDoBlending(Foreground, Background, Alpha, Beta):
56          My_fore = AlphaBlend.DoBlending(Foreground * Alpha, Background * Alpha, Beta)
57          My_back =  Background * (1.0 - Alpha)
58          My_out = My_fore + My_back
59          return My_out
60
```

- My_fore = 前景 * Alpha * Beta + 背景 * Alpha * (1.0 - Beta)
- My_back = 背景 * (1.0 - Alpha)

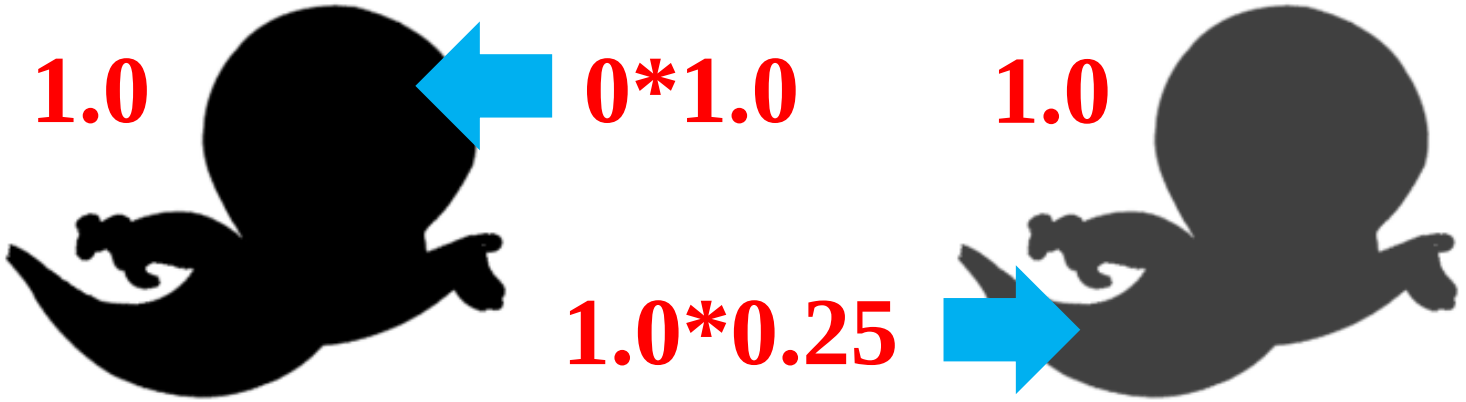
獨特設計之處

- Alpha = alpha, Beta = 1.0
- Alpha = alpha, Beta = 0.75

前景圖

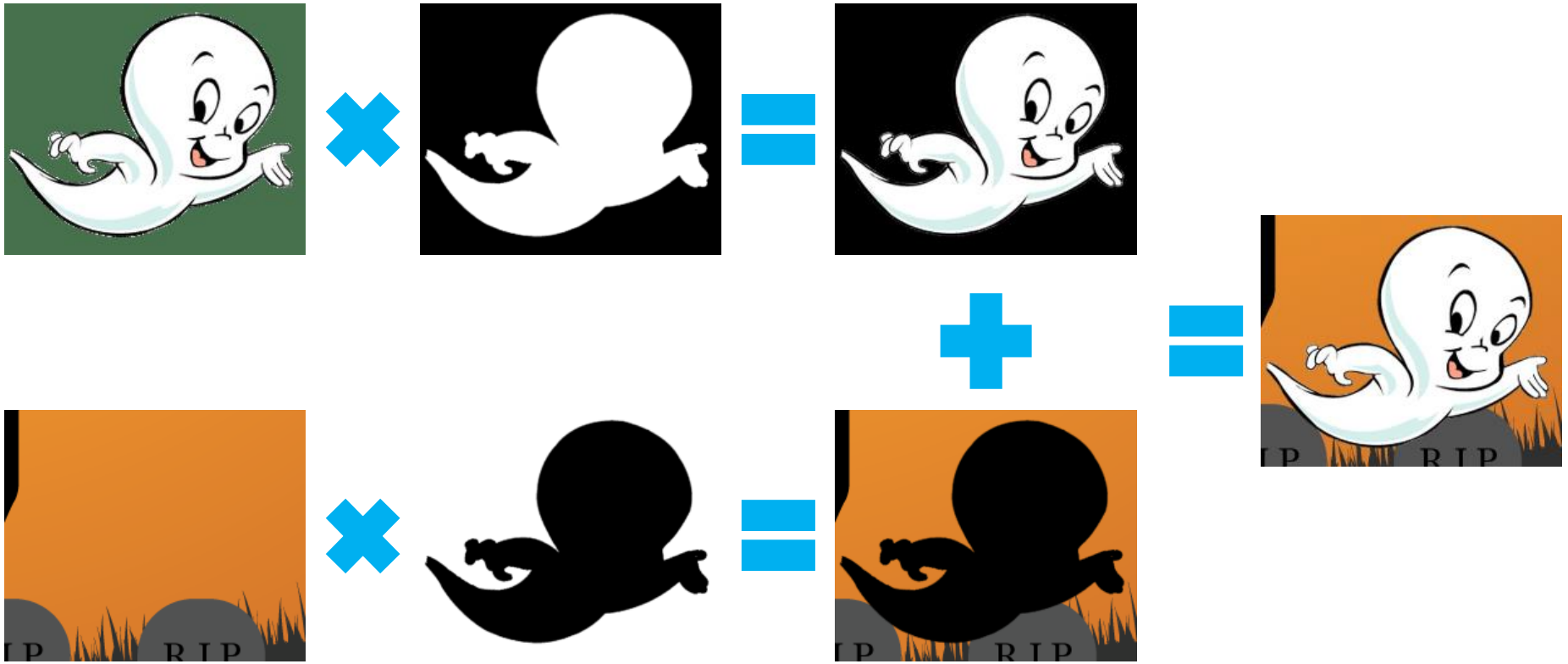


背景圖



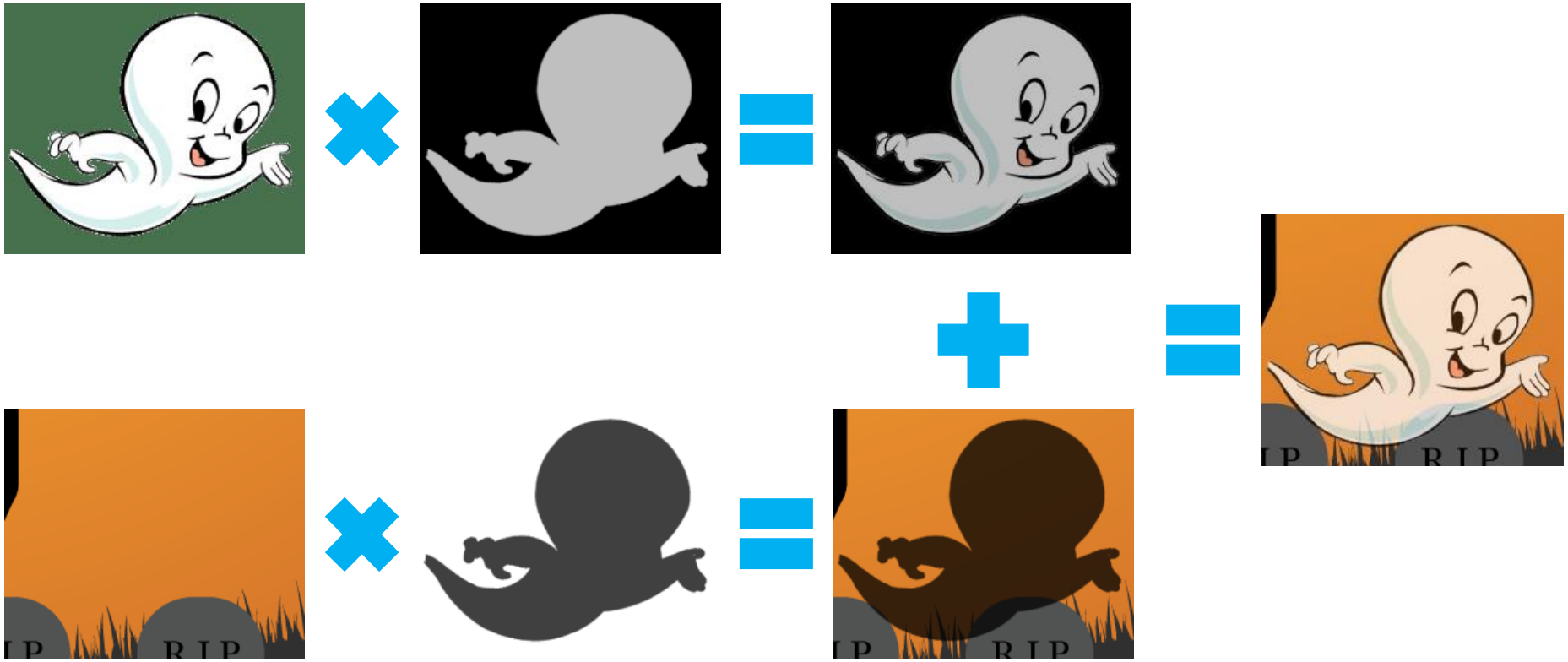
獨特設計之處

- Alpha = alpha, Beta = 1.0



獨特設計之處

- Alpha = alpha, Beta = 0.75



結果展示

組員分工表

組員	工作分配比重	內容
李宗晏	100%	程式、報告

Thanks for your attention