

基於邊緣特性之解拜爾圖像電路設計方法

A Hardware Design of Edge-based Debayering Method

專題組別：2-5

指導教授：陳培殷

發表領域：架構與嵌入式系統計算

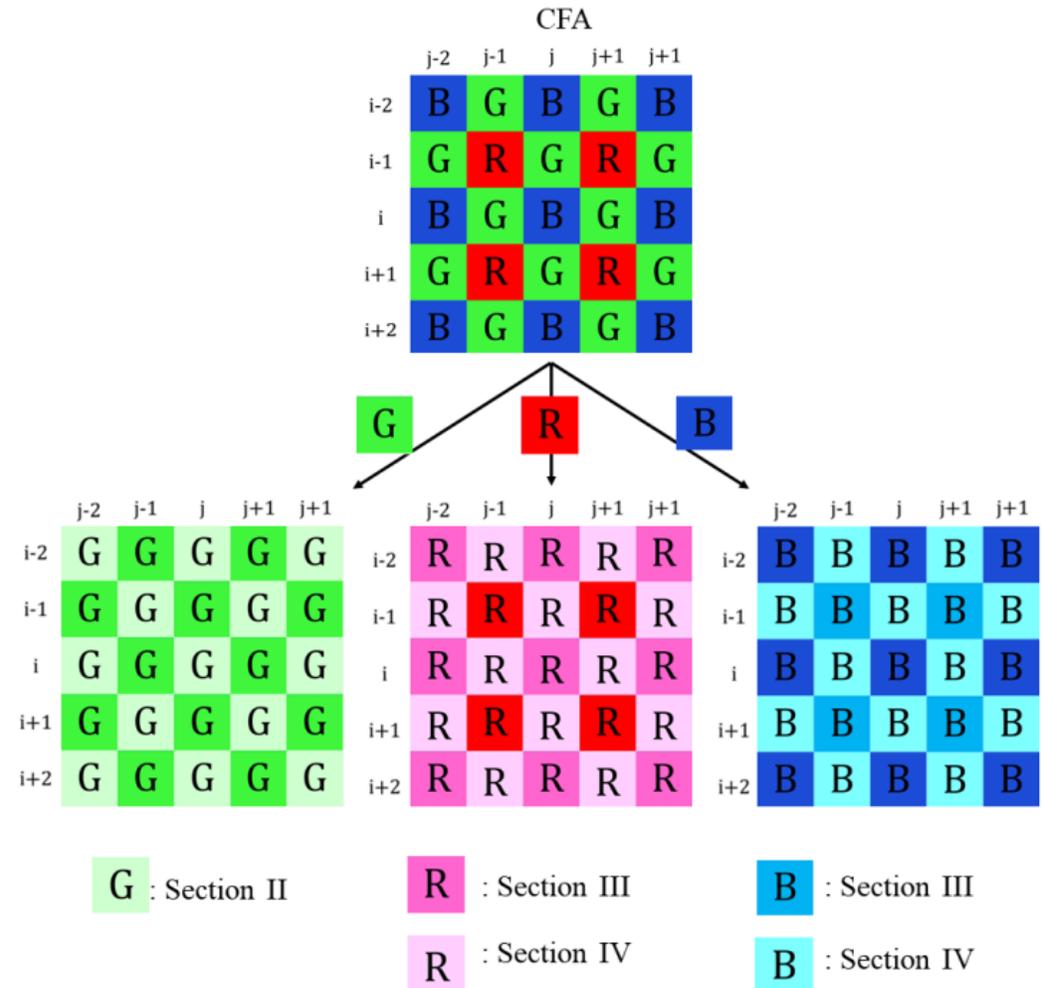
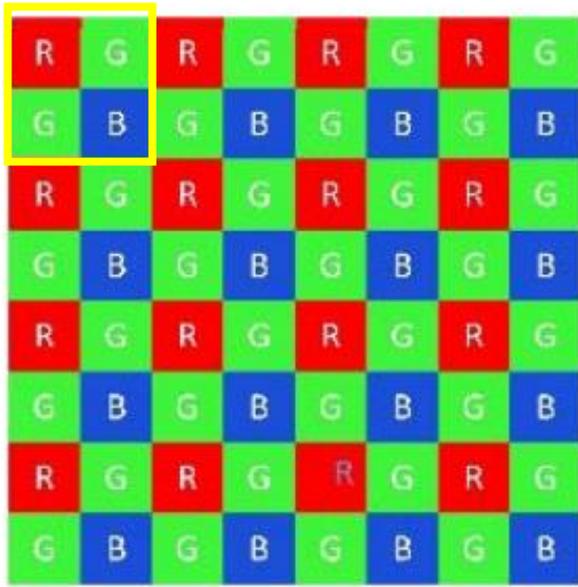
專題成員：杜少宏、葉承軒、楊宇翔、蘇晉德

目錄

- 簡介
- 方法
- 硬體設計
- Demo
- 結論

簡介

將輸入圖片依照color filter array (CFA) 存下對應channel的數值，再按照順序解馬賽克(Debayering) 補齊所有顏色。



方法

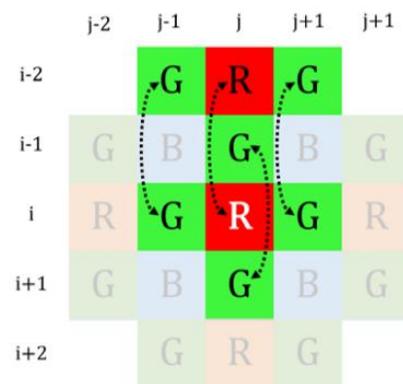
- A Hardware Design of Edge-based Debayering Method
 - > Image size : 400 x 300 pixels
 - > Block size : 5 x 5 pixels
 - > Line buffer: 400 x 7 pixels
 - > 由左至右、由上至下依序掃描圖片並輸出至螢幕

方法：找 G in R

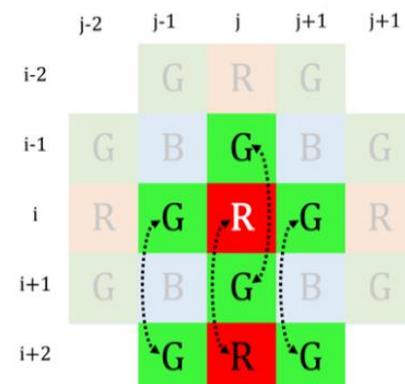
$$Uu(i, j) = \frac{1}{4} \times \begin{pmatrix} |CFA(i, j-1) - CFA(i-2, j-1)| \\ + |CFA(i, j) - CFA(i-2, j)| \\ + |CFA(i, j+1) - CFA(i-2, j+1)| \\ + |CFA(i-1, j) - CFA(i+1, j)| \end{pmatrix} \quad (1)$$

$$D = \frac{1}{Uu(i, j)} + \frac{1}{Ud(i, j)} + \frac{1}{Ul(i, j)} + \frac{1}{Ur(i, j)}$$

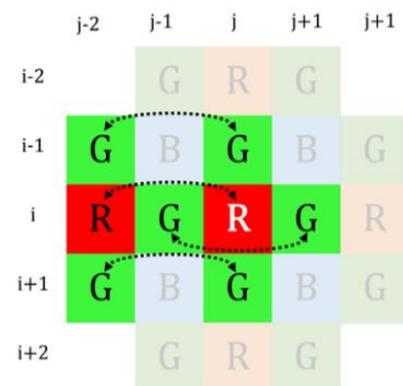
$$Wu(i, j) = \frac{1}{Uu(i, j)} \frac{1}{D}$$



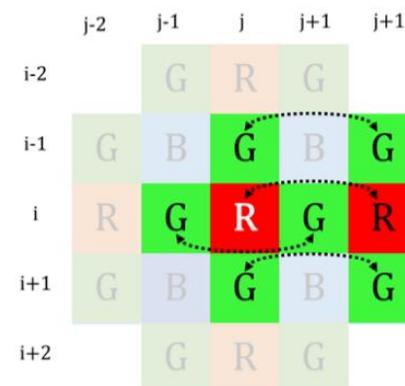
(a)



(b)



(c)



(d)

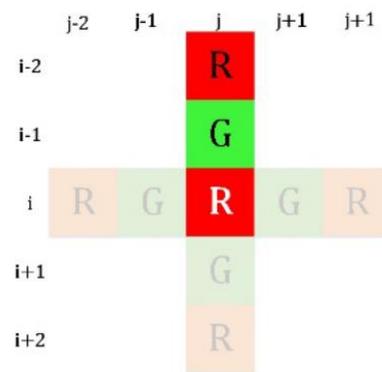
方法：找 G in R

$$\tilde{G}u = CFA(i-1, j)$$

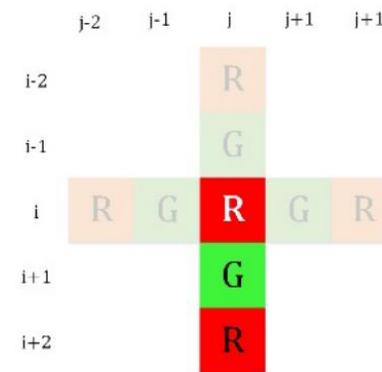
$$\Delta Cu = \frac{1}{2} \times (CFA(i, j) - CFA(i-2, j))$$

$$Gu(i, j) = \tilde{G}u + \Delta Cu$$

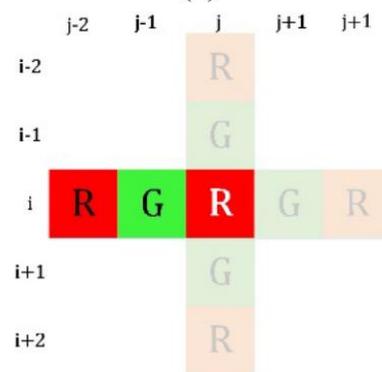
$$G(i, j) = \begin{pmatrix} Gu(i, j) \times Wu(i, j) \\ +Gd(i, j) \times Wd(i, j) \\ +Gl(i, j) \times Wl(i, j) \\ +Gr(i, j) \times Wr(i, j) \end{pmatrix}$$



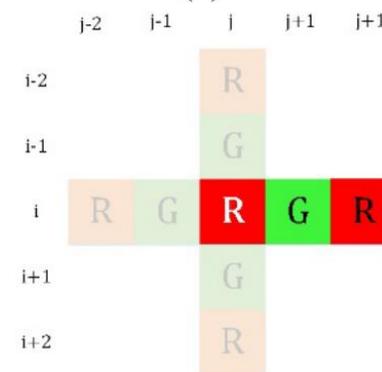
(a)



(b)



(c)

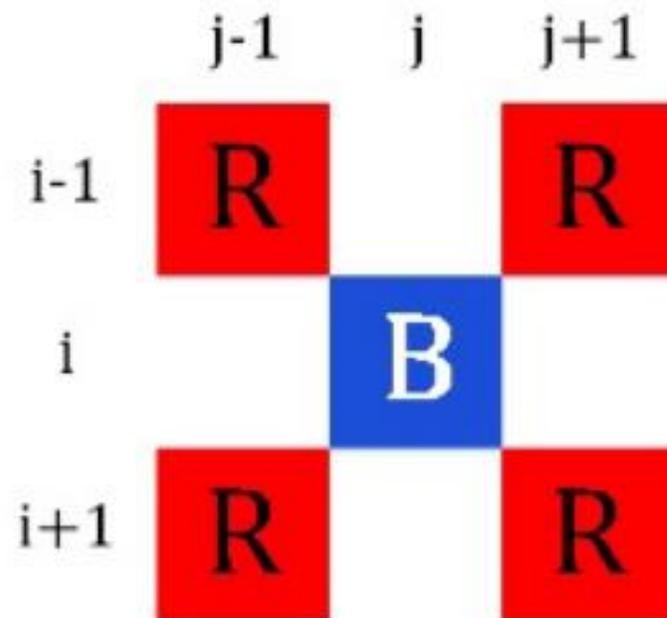


(d)

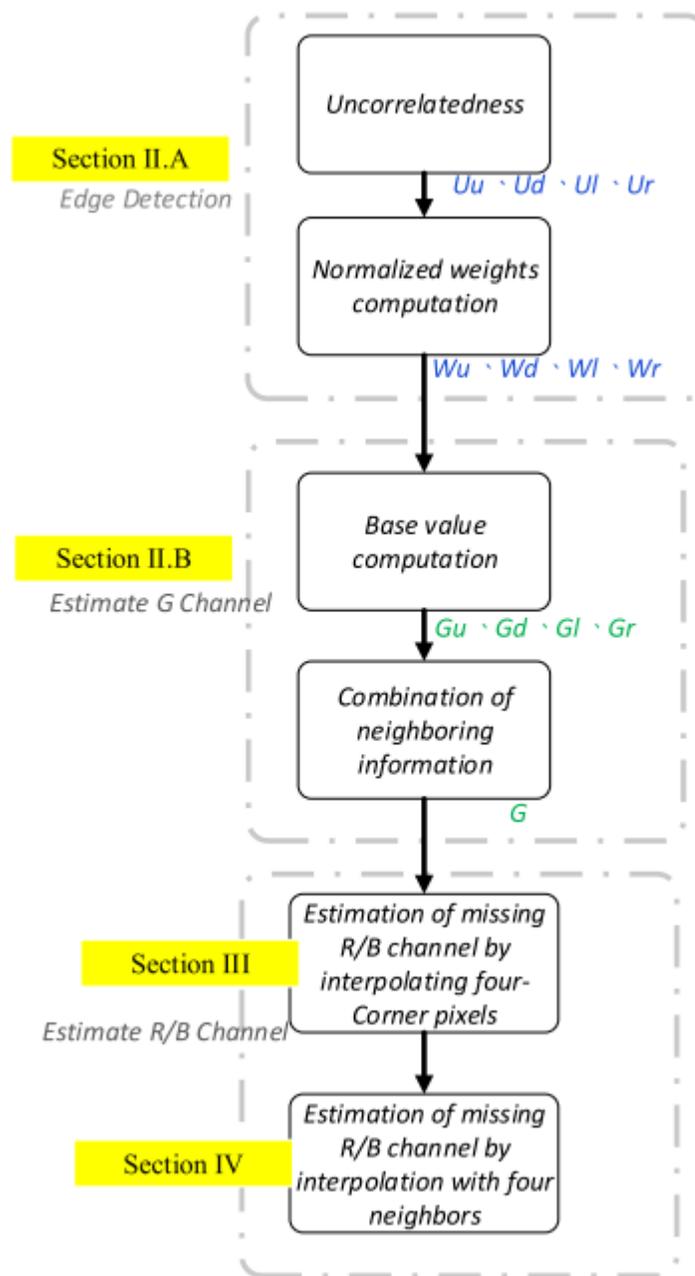
方法：找 B in R

$$d_{BG} = \frac{1}{4} \times \begin{pmatrix} B(i-1, j-1) - G(i-1, j-1) \\ +B(i-1, j+1) - G(i-1, j+1) \\ +B(i+1, j-1) - G(i+1, j-1) \\ +B(i+1, j+1) - G(i+1, j+1) \end{pmatrix}$$

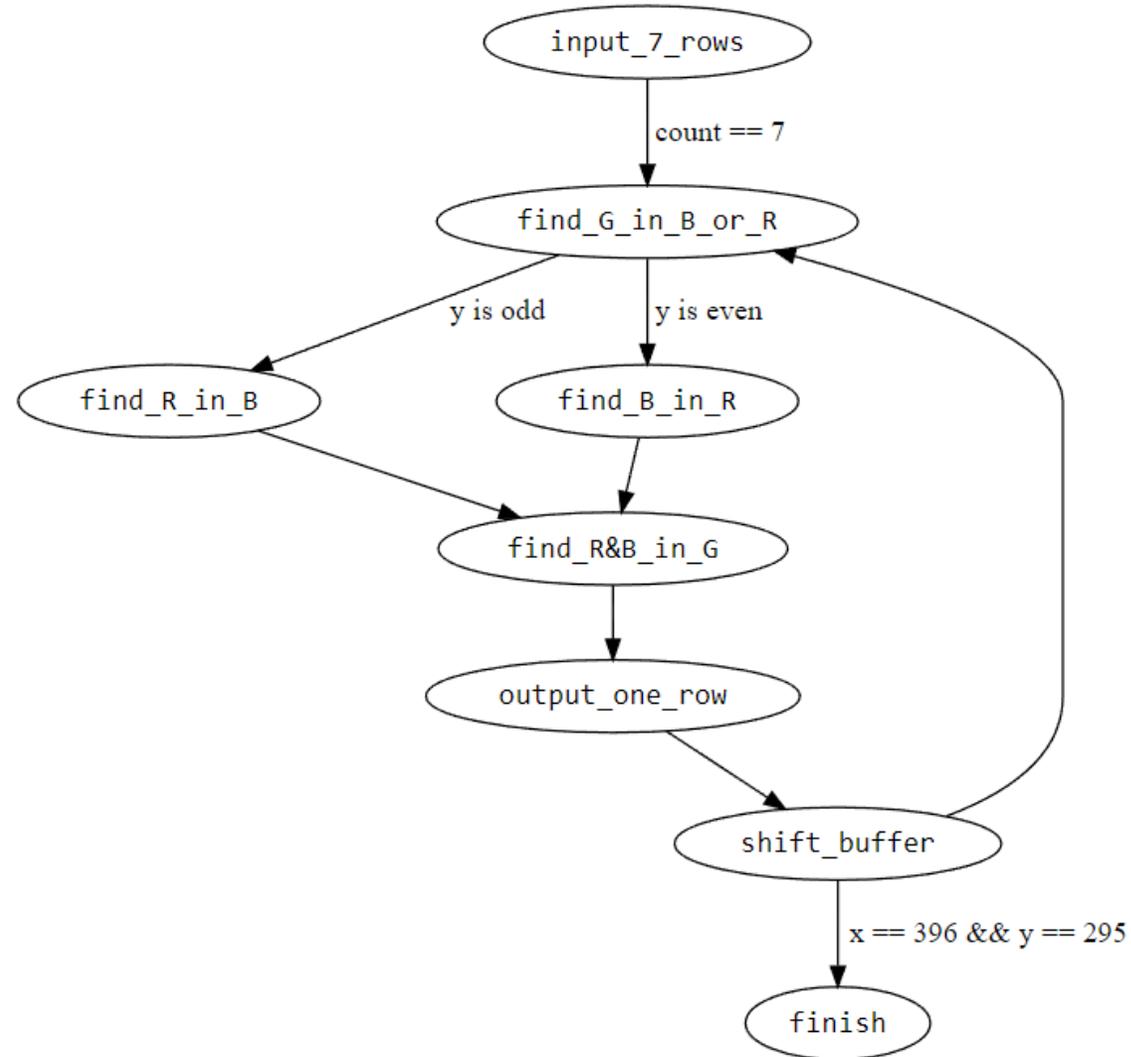
$$B'(i, j) = G(i, j) + d_{BG}$$



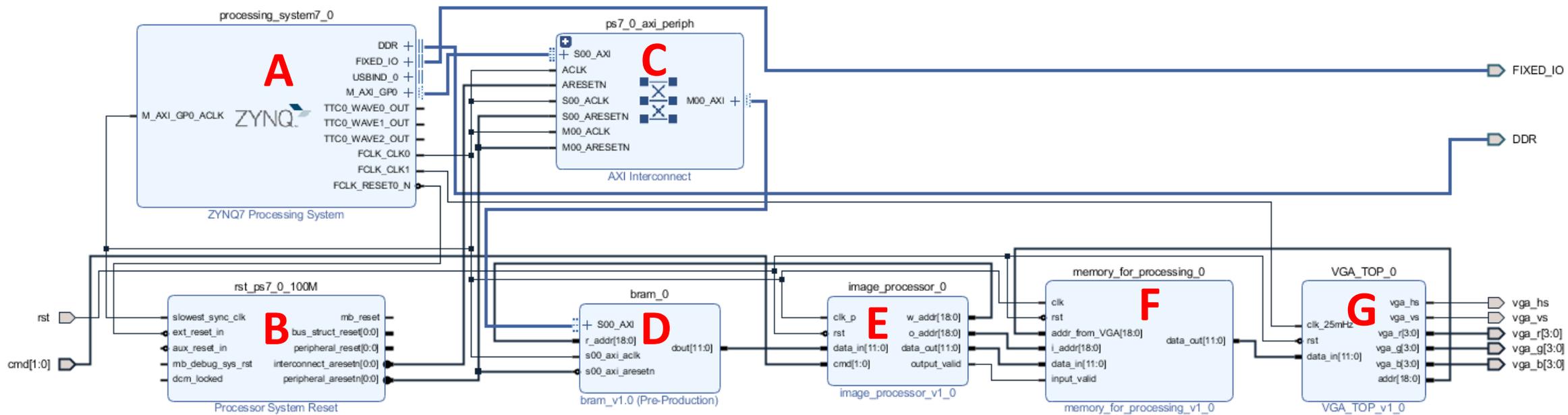
方法：流程图



方法：狀態機



硬體設計



Demo



結論



```
Transcript
File Edit View Bookmarks Window Help
Transcript
# Pixel 118169 is wrong! The output data is 888, but the expected data is 778
# Pixel 118171 is wrong! The output data is 888, but the expected data is 778
# Pixel 118173 is wrong! The output data is 788, but the expected data is 888
# Pixel 118174 is wrong! The output data is 788, but the expected data is 888
# Pixel 118175 is wrong! The output data is 788, but the expected data is 888
# Pixel 118179 is wrong! The output data is 888, but the expected data is 778
# Pixel 118181 is wrong! The output data is 888, but the expected data is 778
# Pixel 118191 is wrong! The output data is 888, but the expected data is 778
# Pixel 118198 is wrong! The output data is 777, but the expected data is 778
# Pixel 118209 is wrong! The output data is 888, but the expected data is 778
# Pixel 118225 is wrong! The output data is 888, but the expected data is 778
# Pixel 118347 is wrong! The output data is 677, but the expected data is 777
# Pixel 118348 is wrong! The output data is 677, but the expected data is 777
# Pixel 118349 is wrong! The output data is 677, but the expected data is 777
# Pixel 118350 is wrong! The output data is 677, but the expected data is 777
# Pixel 118351 is wrong! The output data is 677, but the expected data is 777
# Pixel 118352 is wrong! The output data is 677, but the expected data is 777
# Pixel 118353 is wrong! The output data is 677, but the expected data is 777
# Pixel 118354 is wrong! The output data is 677, but the expected data is 777
# Pixel 118355 is wrong! The output data is 677, but the expected data is 777
# Pixel 118356 is wrong! The output data is 677, but the expected data is 777
# Pixel 118357 is wrong! The output data is 677, but the expected data is 777
# Pixel 118358 is wrong! The output data is 677, but the expected data is 777
# Pixel 118359 is wrong! The output data is 677, but the expected data is 777
# Pixel 118360 is wrong! The output data is 677, but the expected data is 777
# Pixel 118361 is wrong! The output data is 677, but the expected data is 777
# Pixel 118362 is wrong! The output data is 677, but the expected data is 777
# Pixel 118363 is wrong! The output data is 677, but the expected data is 777
# Pixel 118364 is wrong! The output data is 677, but the expected data is 777
# Pixel 118365 is wrong! The output data is 677, but the expected data is 777
# Pixel 118373 is wrong! The output data is 556, but the expected data is 666
# Pixel 118375 is wrong! The output data is 556, but the expected data is 666
# Pixel 118397 is wrong! The output data is 060, but the expected data is 966
# Pixel 118398 is wrong! The output data is xxx, but the expected data is 060
# Pixel 118399 is wrong! The output data is xxx, but the expected data is 006
#
# ----- S U M M A R Y -----
#
# There are      17429 errors !
#
#
# ** Note: $finish      : C:/Users/user/Desktop/tb/testfixture.v(101)
# Time: 7651080 ns  Iteration: 0  Instance: /test
```

約12萬筆資料中只有17429有誤差，且影像還原度為94%~100%

結論



原圖



演算法完成圖

END