Screen Burn-In Correction Program

2016550027 이태규

I Introduction

1 Raising a Problem: Display Problems and Difficulty of Solving

- Recently, Samsung and Apple introduced a display capable of 1200Hz refresh rate on smartphones, attracting people's attention. However, not long after, some problems such as 'odd green tint' on the screen are reported by a large number of users. Manufacturers have also identified problems and tried to solve them through system updates, but some users still experience 'odd green tint' problem. It may because the degree of the problem is different for each user's device, and the update initiator cannot check it one by one.
  This is not the only situation for the 'odd green tint' problem. In a display using an OLED panel, burn-in can easily occur, which is an inevitable problem due to the characteristics of the manufacturing process, and the degree of the problem is bound to vary from device to device. In addition, even when the same screen is displayed for a long time on an OLED display, a burn-in phenomenon occurs. Until now, hardware repairs were required to solve this problem. However, this cost both time and money. So, what if the user could solve the screen burn-in problem through a simple program?

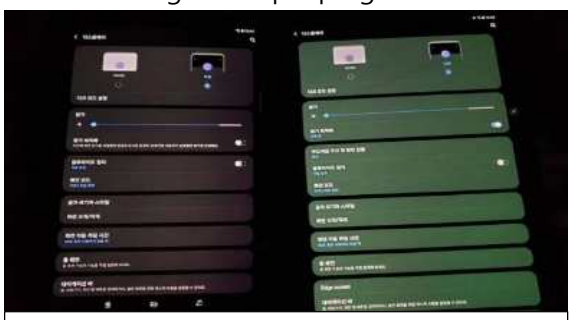


figure 1. 'odd green tint' problem in smart phone

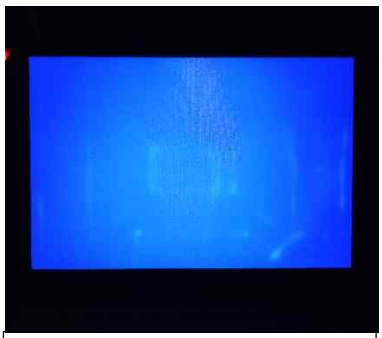


figure 2. 'burn-in' problem in my lab-top

II Suggest Solution:

- Using the image data of the screen where the burn-in phenomenon has occurred, the brightness of the display domain is matrixed and the image is corrected so that the burn-in is not visible on the screen..

III Step by step goal:

1 Create PNG image data by taking a picture by displaying a blue, red, green screen on the burn-in display.

2 The red, green, and blue data of the generated image are separated, and the data is organized by classifying the domain where the problem occurs due to burn-in from the data and the difference in brightness that occurs during shooting.

3 Create a matrix so that the brightness is the same in all domains.

4 Create a new r, g, b image using the created matrix, and check if this image can correct the actual screen.

5 If the r,g,b correction is successful, apply it to any photo.

IV Preliminary investigation

1 Can the cell data of png files be handled in python?

- Yes
  Using the PIL module, I can load png files from python and store pixel data in a numpy array.

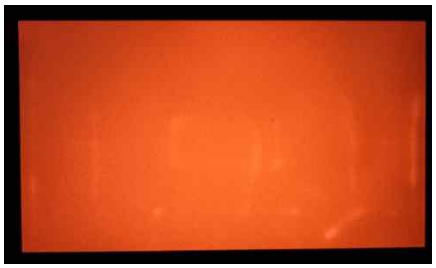
```
import numpy as np
from PIL import Image

img_name = ('my_image.jpg')
im = Image.open('python_img\\'+img_name)

pix = np.array(im)
```

2 Is it possible to get the data corresponding to the burn-in phenomenon from the image obtained by actually shooting the burn-in phenomenon?

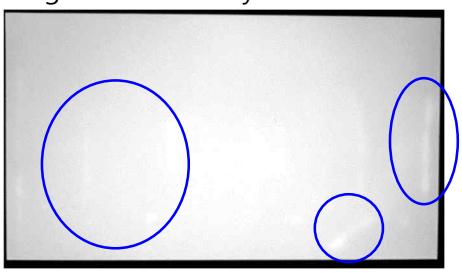
- Yes
  Using the program,'Igor', I checked how the burn-in domain is classified in the data.
  Depending on the photographing method, some burn-in domains may not be identified, but it could be easily overcome by adjusting the brightness.

B Data verified by Igor

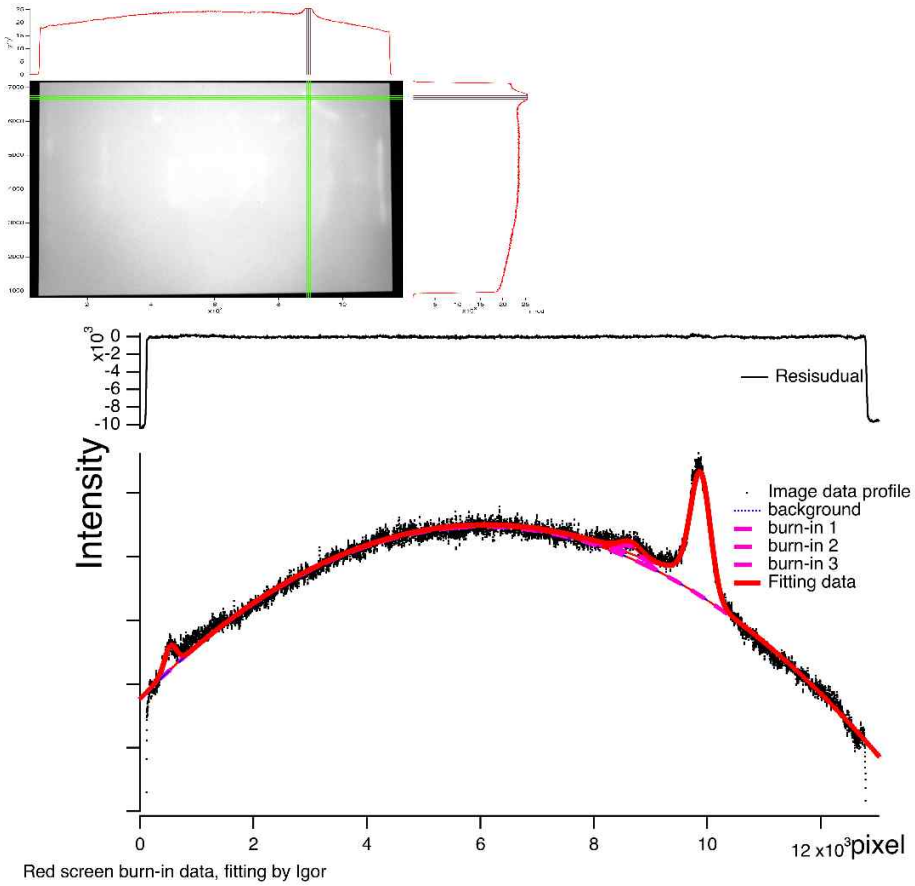
- Original image



- Image extracted only red data



- Profile of Intensity vs pixel by Igor



Red screen burn-in data, fitting by Igor

3 burn-in domain is verified

V Materials & Module, I need

1 Display with burn-in

- I have a laptop I've been using for 5 years, and its display have 'burn-in' domains.

2 Good camera

- I will use my smart phone camera. but if I need better camera, I will find others.

3 PIL module

- Maybe I should ask to Professor