Team08_final_report

Image Dehazing Accelerator

組員：陳昱仁、楊欣哲、傅敬倫

1. Motivation

將有霧的照片去霧，並相較於軟體有達到加速的效果

2. Dehaze Algorithm

- Dark channel prior : In most fog-free photos, every pixels exists at least one channel has very low values

    1. Dark Channel can be represented by :

    $$J^{dark}(x) = \min_{y\in\Omega(x)}\{\min_{c\in r,g,b}(J^c(y))\},$$

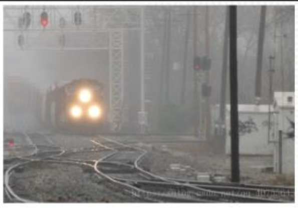
    $$J^{dark}(x) \rightarrow 0 \text{ (fog-free photos)}$$

    2. $\Omega(x)$ is a smooth filter, we use Blur filter :

$$\frac{1}{kernal\ size^2}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Filter

Following is Dark channel of some fog and no fog picture :

	Original	Dark channel
Fog picture		
No Fog picture		

We can see the dark channel of fog picture isn't black. And the dark channel of no fog picture is black. This example shows that dark channel prior is correct.

- By Kaiming He defog model :

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})\mathbf{t}(\mathbf{x}) + \mathbf{A}[\mathbf{1} - \mathbf{t}(\mathbf{x})]$$

I(x) is origin picture (including fog)

J(x) is defog picture (wanted)

t(x) is transmittance (constant)

A is atmospheric light (constant)

- By above two theorem, we can get no fog picture $J(x)$
- we need first compute $t(x)$:

Computing $t(x)$

$$I(x) = J(x)t(x) + A[1 - t(x)] \cdots (1)$$

$$\frac{I(x)}{A} = \frac{J(x)}{A} t(x) + [1 - t(x)] \cdots (2)$$

$$min_{y\in\Omega(x)}\left\{min_{c\in r,g,b}\frac{I(x)}{A}\right\} = min_{y\in\Omega(x)}\left\{min_{c\in r,g,b}\frac{J(x)}{A}t(x)\right\} + 1 - t(x) \cdots (3)$$

$$J(x) \text{ is defog picture, } min_{y\in\Omega(x)}\left\{min_{c\in r,g,b}\frac{J(x)}{A}t(x)\right\} \to 0$$

$$\therefore t(x) = 1 - min_{y\in\Omega(x)}\left\{min_{c\in r,g,b}\frac{I(x)}{A}\right\}$$

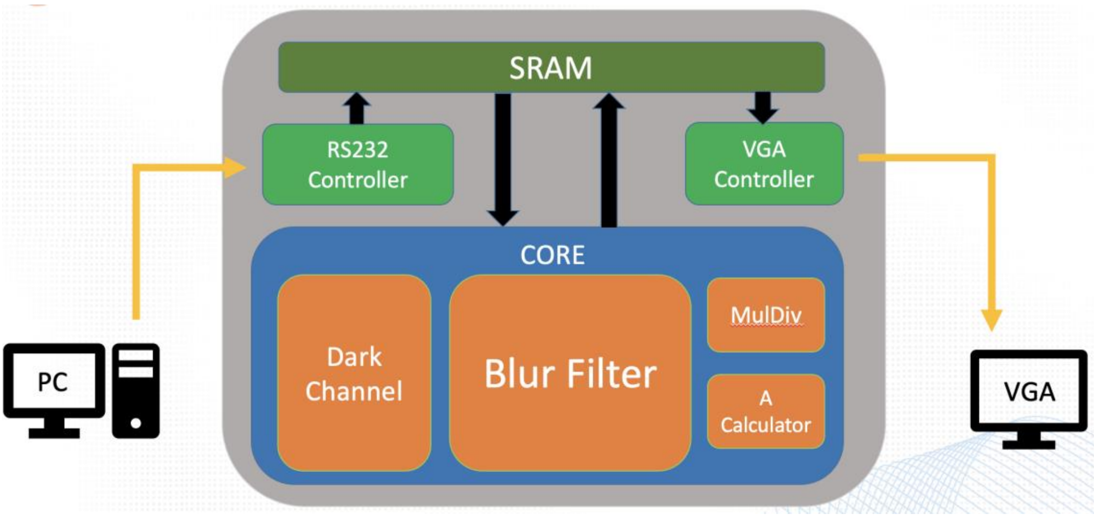
- Define $P^{dark} = min_{c\in r,g,b} I$, $J(x)$ can be denoted as :

$$\mathbf{J(x)} = \frac{\mathbf{I(x) - min_{y\in\Omega(x)} P^{dark}}}{\mathbf{1 - min_{y\in\Omega(x)} \frac{P^{dark}}{A}}}$$

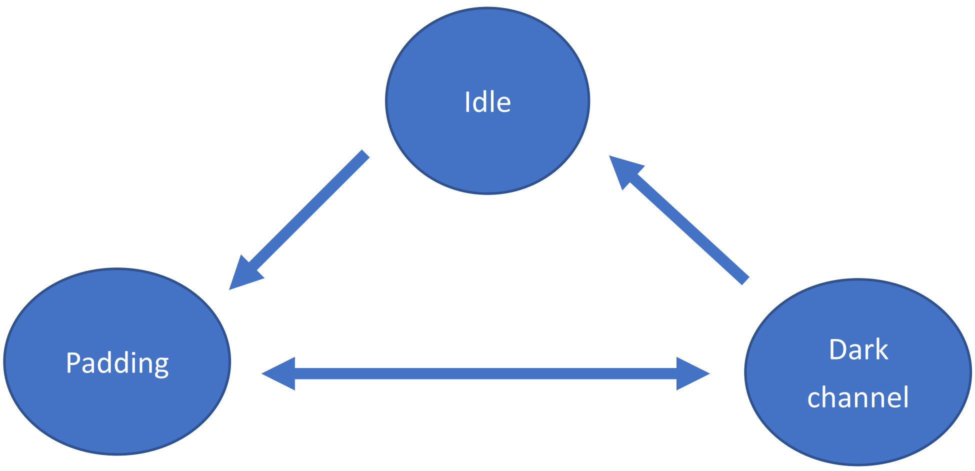
- Our procedure

```
I(C) is original picture, we reshape to 3*480*320
Compute Dark channel : min(r, g, b)
Do convolution with Dark channel by blur, denoted as BD
Limit the Dark channel's range : min(Dark * w, 204), w = 0.96875
Compute A
Compute Defog picture : J[:,:,C] = [I(C) - BD]/(1 - BD/A)
```

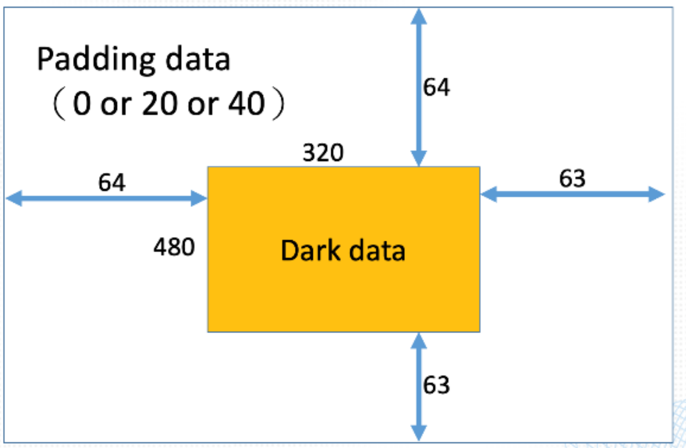
3. Hardware block diagram



■ Dark channel module:



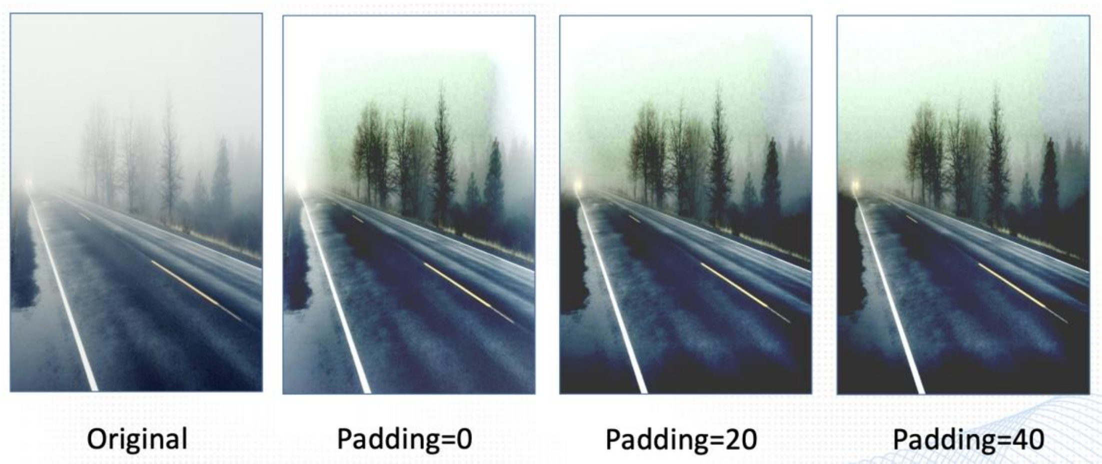
Data location in sram:



我們希望在Dark channel module將padding data and dark data 依照上圖的地址存入sram,方便Blur module 取得正確的資料·

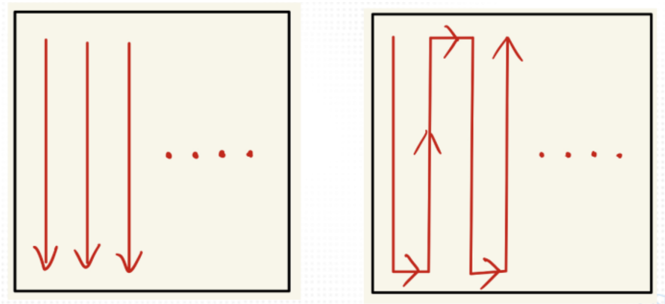
Sram 的地址照順序是由左到右，由上而下

■ **Padding**



對於 convolution，我們採取 constant padding·起初我們採取 zero padding，但會導致邊界的 dark channel 的值過小，進而導致邊界的值過大，產生邊界過白的現象。我們發現將 padding 的大小調到 40 可以去除邊界過白的現象·

■ Blur



在進行 blur 運算，我們希望可以最大利用ＳＲＡＭ的資料，因為若每個 pixel 都重新 load 資料，每次都要花 128*128 個 cycles。

我們設計了兩套資料存取得流程，第一套的存取順序如左圖，在每行的第一個 pixel 皆花 128*128 個 cycles 將所需資料 load 進來，接下來得其他 pixel，因資料是相鄰，只需多花 128 cycles 將下面一列的資料傳進來，即可完成運算。

第二套存取順序如右圖，由第一套存取流程改良而來，第一套流程的缺點是在每行第一個 pixel 皆須不必要的等待，我們發現除了第一行的第一個 pixel 需等待 128*128 個 cycles，其他的 pixel 的等待時間可以透過 s 型存取順序來避免等待過長的問題，最後我們是使用第二套存取流程。

4. Hardware efficiency

Software（python）: 2.4s

Hardware :

- Frequency : 75M
- Image size : 480 * 320
- Time : 0.5s

5x speed up

5. Problems

- Rs232 spent too much time
- Blur’s kernel size is too big（128*128）, which is bad for timing performance
- Other padding methods may have better result

6. reference

Single Image Haze Removal Using Dark Channel Prior, by Kaiming He https://ieeexplore.ieee.org/document/5567108