**GPU Lab3 Report**

R04943015 高聖鈞

**Acceleration**

我採用助教所建議的coarse to fine的方式做加速，以不同scale1/8x、1/4x、1/2x和1倍，分段去做處理，如此，iteration數可以大幅下降，以下是這次作業的分析。

* **1/8x scale**
  + 因為這次的複雜度下降是O()，所以長寬減為八分之一時，所需要的計算量，會降為六十四分之一。
  + 若以助教所提供20000次iteration為可接受收斂值，則20000/64=312.5，這個結果，應該可以在313次後收斂。圖一為這次的結果。
  + 由圖一可知，在313次iteration後，確實收斂了，iteration再變多，也不會有顯著變化。可以看到圖上格子比較大，那是因為現在較為coarse。另一個問題是，邊界會有些白邊，那是因為我在處理縮圖時，是以較基本的方式處理，直接以整除八的方式縮圖以及放大回來，所以本來對於1/8x來說邊邊的小白點，會被放大64倍。
  + 但由於下個scale的iteration會更為精細，就可以逐漸把這白邊補齊。



*(圖一)*

* 1/4x
  + 依照前面的推倒這次的iteration數應為20000/16=1250次，其結果如圖二，圖二的結果為沒有參考1/8x的結果，直接從頭開始iteration，可以看到也如預期收斂，而且白邊也變小了。



*(圖二)*

* + 這次我們讓1/4x接著，1/8x收斂後的結果做，一樣做1250次iteration，得到結果圖三。可以看到和圖二效果差不多，然而因為在1/8x時，已經有coarse的answer，所以可以不用那麼多次iteration。



*(圖三)*

* + 因此和上面情況相同，但我們選擇iteration數較少的300次。結果如圖四。也已經足夠收斂。



*(圖四)*

* + 由以上討論可以看出coarse to fine strategy 對於效能提升的幫助。
* 1/2x,1x scale
  + 基於以上的討論，知道每次的iteration數都可以因為承接之前的結果，這邊我們對於1/2x取iteration數1250，1x取iteration數5000，其結果如圖五。可以看到已達到收斂。



(圖五)

**Discussion**

* 原先需要20000個iteration，而經過這樣的改善可以大幅下降iteration次數，那可以降低的幅度是多少。
  + 我的想法是，第一次iteration:1/8x必須讓他足夠收斂，因為在這邊收斂的答案對後面iteration次數的減少最有幫助，所以我讓1/8x維持313次iteration。
  + 而之後幾個scale的iteration，主要考量是，要讓邊界的顏色足夠影響到整張圖，因此我參考了target的影像大小為384x216，所以對1x scale的圖，我會做大約(384+216=)600次iteration，而1/2x則做300次，1/4x做150次。依照如此的推論設定，可以得到圖六。觀察圖六的結果也已經達到收斂。



*(圖六)*

**Conclusion**

採用coarse to fine strategy的方式達到加速效果，讓原先20000次收斂的計算降到大約1360次，降幅為93.2%，效果相當顯著。