

# Final Project Report

## Sky Replacement

Team 11

組員：王領崧 陳伯瑾 朱偉綸

# Outline

- Motivation
- Methods
- Improve After Demo
- Results & Analysis
- Conclusion
- Discussion
- Reference

# Movtivation

老師提供的Term Project主題中，Sky Replacement的Paper 深深吸引住我們的目光，不單單是因為結果呈現的視覺效果很好，仔細看置換天空後的地面細節，光影的變化，轉換的也十分擬真，彷彿場景真實存在般。因此，秉著強大的好奇心，稍微讀過了整篇Paper，發現其sky segmentation與replacement的部分，都是使用了FCN進行操作，再查看了其他幾篇類似的paper後，發現所用的手法大同小異，似乎都跟AI脫不了關係，雖然這讓我們有點小小的失望，但同時也激起了我們的決心，想要運用上課所學到的知識，擺脫AI的方式，也能得到這麼好的效果。

## Sky is Not the Limit: Semantic-Aware Sky Replacement

Yi-Hsuan Tsai<sup>1\*</sup>

Xiaohui Shen<sup>2</sup>

Zhe Lin<sup>2</sup>

Kalyan Sunkavalli<sup>2</sup>

Ming-Hsuan Yang<sup>1</sup>

<sup>1</sup>University of California, Merced

<sup>2</sup>Adobe Research



Input photograph

Automatic sky replacement results with different styles

# Methods

實作的方法主要分成以下五個步驟: (此頁僅會簡單的稍作說明，詳細講解會在後面幾頁分別介紹)

1. Sky segmentation (將input的圖片分割成地面與天空兩個部分)
2. Retrieve sky images (從sky dataset 取出要置換的天空背景)
3. Realistic sky style transfer (利用欲置換的天空對input的地面部分進行風格轉換)
4. Sky replacement (將input的天空替換為要置換的天空)
5. Boundary smoothing (對新圖片中天空與地方的邊界進行smoothing的動作)

# Step1 : Sky Segmentation

起初因為對sky segmentation沒什麼想法，因此參考某篇paper based on border points的algorithm。做法主要可以拆成以下幾個步驟：

1. 計算整張圖片的gradient
2. 設置一個threshold
3. 跑過input圖片的每一個column，紀錄每個column從上到下，第一個發生gradient > threshold的位置
4. 將每個column的點做連接，得出天空與地面的界線，將其丟入Energy function做運算
5. 紀錄energy function的計算結果，回到步驟2並重新設置一個threshold，經過多次的調整與運算後，得出optimal(max) Energy function，即為最佳的天空與地面的界線

下圖為簡易的pseudo code和Energy function的計算公式

---

```
Algorithm 1: Calculate sky border position function:  
Calculate_border(grad, t).  
Input: threshold t; gradient image grad.  
Output: sky border position function b (x)  
-----  
for x = 1 to W  
    b(x) = H  
    for y = 1 to H  
        if grad(y,x) > t  
            b(x) = y  
            break  
        end if  
    end for  
end for
```

---

$$J_n = \frac{1}{\gamma \cdot |\Sigma_s| + |\Sigma_g| + \gamma \cdot |\lambda_1^s| + |\lambda_1^g|}$$

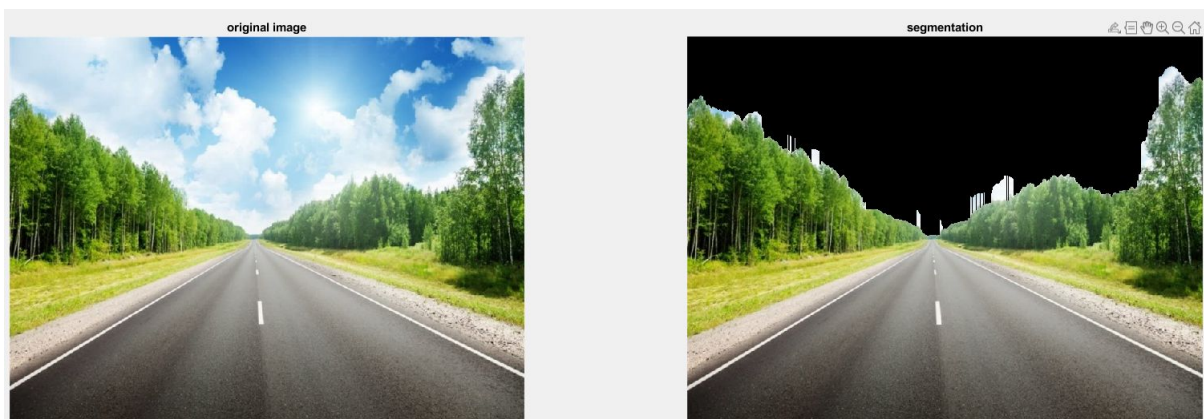
然而，在實作的過程中，我們發現每輪計算Energy function的過程中，都要跑過整張圖片數次，並且調整threshold的數量亦不能太少，整個過程就如同窮舉的方式(暴力展開)，計算量太過龐大，導致我們跑個10幾20分鐘左右都還沒有結果。再者是利用第一個發生gradient > threshold的位置，這個方法有失一般性，因為很多圖片的天空可能會有一些雲層的交疊，很容易就會使得程式誤判，而且也有可能這張圖片界線的部分gradient數值不一致，導致用此種方法根本找不出最佳解。

因此後面改使用上課所學的superpixel，利用recursive superpixel(在每次superpixel過後，都會將同樣區塊的顏色統一設為該區塊的mean)

來使得圖片的pixel總數不斷的下降，希望能藉此一路分到2塊(即為天空與地面)，不過我們發現當圖片的pixel數量越少時，礙於superpixel為local comparison的演算法，在分類的過程中，很容易因為相鄰顏色過於相近而被汙染，導致結果的界線都不太精準，儘管將最後幾次的superpixel改為global comparison的Kmeans，試圖解決上述的問題，然而結果仍然不盡理想，界線也都稜稜角角。



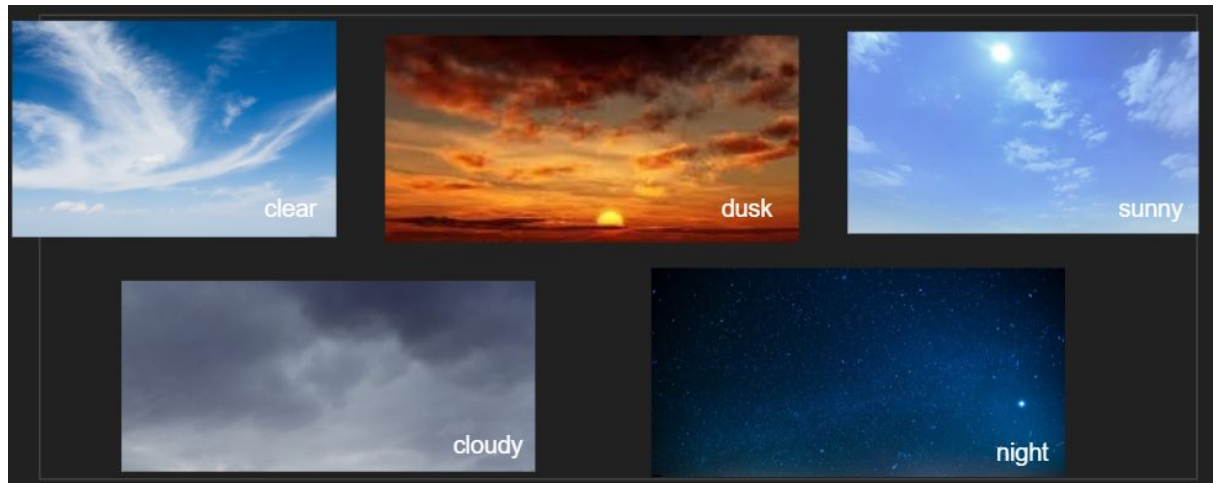
所以，最後結合了上述兩種方法優勢的地方，採用了superpixel + HSV\_threshold的方式，仍然先使用superpixel來降低pixel的總數量，等降到一定數量後，設置一個hsv\_threshold，並且依序跑過圖片的每一個column，把藍白色與非藍白色的交界點記錄下來，連接所有column的界線即為結果。另外為了使分類的效果更加穩定且良好，我們將input的圖片規範為山景圖片。



## Step2 : Retrieve Sky Image

為了簡化選取Sky image的動作(非Project的重點)，我們挑選五種常見的天氣狀況做為dataset，對每張input image進行五種風格上的轉換。

下圖由上至下，由左至右依序為clear(無雲、無太陽)、dusk(黃昏)、sunny(艷陽高照)、cloudy(陰天)、night(有些許星空的夜晚)





## Step3 : Realistic Sky Style Transfer

關於Realistic Sky Style Transfer的部分，初次看到其結果時，由於黃昏的Sky Style Transfer，相較於其他時刻或天氣而言，呈現的結果較為直觀，能一眼就看出黃昏時，昏黃的色調映照在大海上的模樣，因此自然而然地認為，其實Sky Style Transfer，就是一種Color Transfer的使用，所以就參照了講義上有提到的Reinhard實作。

首先，我們定義想要換掉天空的風景圖為source image，想要換成的天空圖為target image。

### Method 1 : Reinhard (mod.)

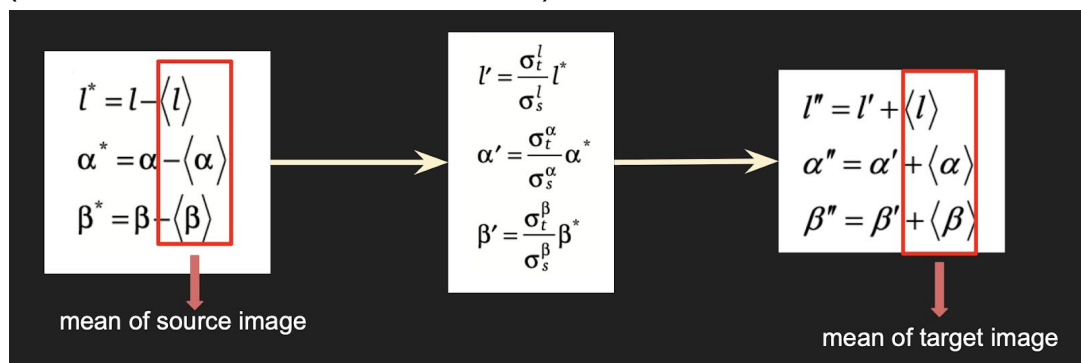
Reinhard的步驟為：

Step1: 將color space從rgb space轉到lab space

Step2: 在lab的space分別計算source image以及target image在l, a, b channel各自的mean和standard deviation，接下來，對source image上的每一個pixel，取其l channel值，減去source image在l channel的mean，再除以source image在l channel的standard deviation並乘上target image在l channel的standard deviation，最後再加上target image在l channel的mean。

此流程也分別對a channel以及b channel實作。如此運算的目的是希望source image在色調上的數值較趨近於target image，且令source image中pixel間的色差分佈較趨近於target image。

(可以見下圖公式運作，較為清楚)



Step3: 再將color space從lab space轉回rgb space

之所以說是Reinhard的改版是因為我們想說自己刻rgb轉lab和lab轉rgb感覺會使轉換速度降低，所以起初在color space的轉換上，是直接呼叫matlab的內建function。

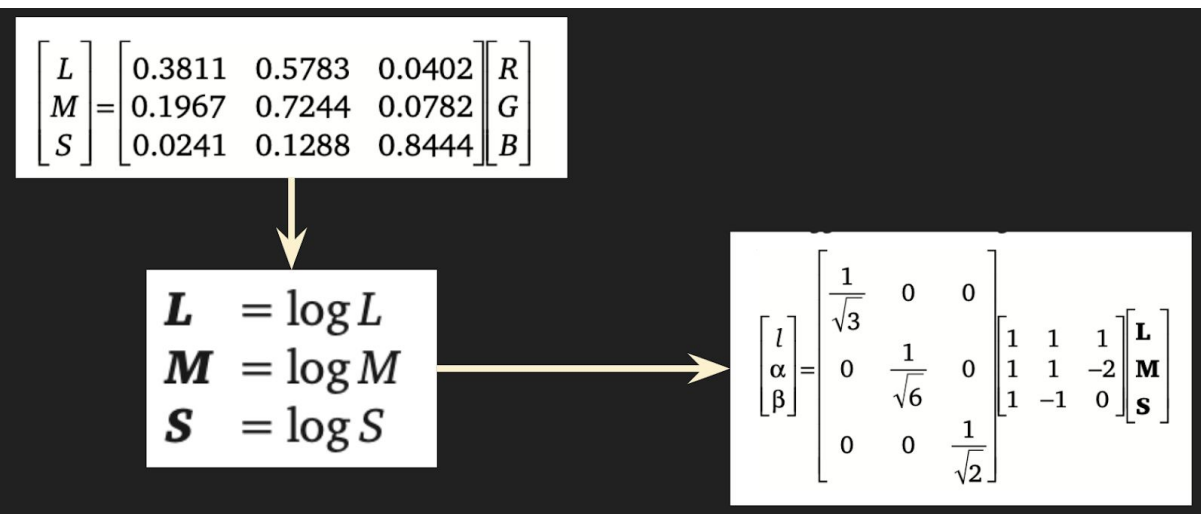
而在實作此method上，我們感覺呈現出來的結果不太滿意，有些轉換的結果圖片，景觀部分會有一種像是蓋上天空圖色調的玻璃紙，再看圖片的感覺，圖片整理上會有霧霧的感覺，如下圖，下方的草景，就如同蓋上一層灰色的玻璃紙一樣，質地間訊息變得相較原圖(右圖)不明顯許多，因此我們想說，會不會matlab內建的轉換function和ReinHard在paper中實作的轉換方法有所出入，於是我們就嘗試自己照著Reinhard的方式，手刻color space的轉換。



## Method 2 : Reinhard

Reinhard步驟在前一個method已有提過，而相較前面不同的是，color space的轉換，我們是照著Reinhard的轉換方式實作。

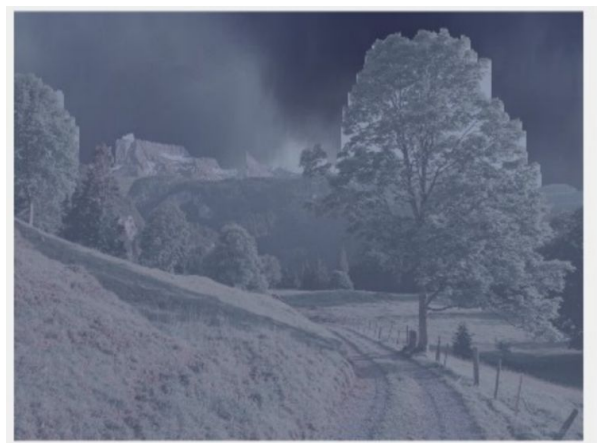
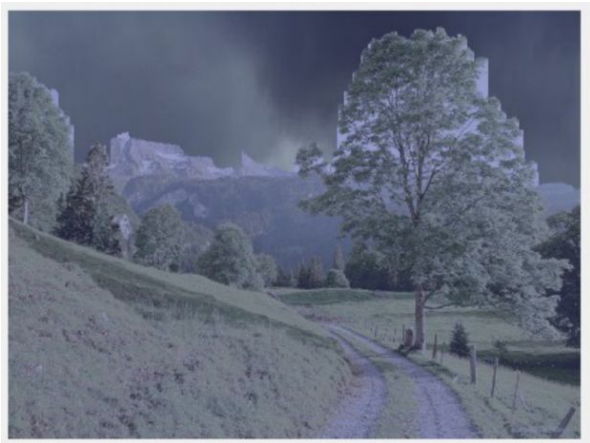
Step1: 將rgb轉lab的部步驟，首先將RGB三個channel乘上定義好的3\*3矩陣，得到LMS，接著對LMS取log，最後再乘上另外兩個定義好的3\*3矩陣，得到lab。





Step3 : 將lab轉rgb的部步驟，將Step2求出來的轉換過後的lab，根據Step1轉換時做的矩陣運算及function操作，逆向做反矩陣與inverse function運算。

而使用Reinhard的轉換方式(左圖)後，相較於method 1用內建function轉換(右圖)，單就此組source image及target image來說，霧化感覺比較沒有那麼嚴重了，質地上感覺也保留了較多的訊息。



但若是其他組source image及target image，用Reinhard的轉換方式也未必較好，如以下組合，感覺用Reinhard的轉換方式，轉換出來的草景就紅的很不自然，在這組source image及target image的case下，感覺用內建function結果就比較自然。(見以下四張圖)



source image

target image



Reinhard轉換



內建function轉換

另外，此兩種method都無法應付轉換成藍天的case，Reinhard(左圖)會轉出不自然的色調，或在其他source image，target image的組合中，如matlab內建function轉換(右圖)一般，會轉出一層藍藍的色調，照在景觀上。



Reinhard轉換



內建function轉換

於是我們想了一下，實際上，sky style transfer不完全是color transformation，假如是轉換成黃昏或晚霞的天空，確實需要做color transformation將黃昏或晚霞的色調投射到地景上，但如果是像轉換成藍天的話，其實應該對地景做的處理是提高亮度，而非將藍天的色調投射到地景上。

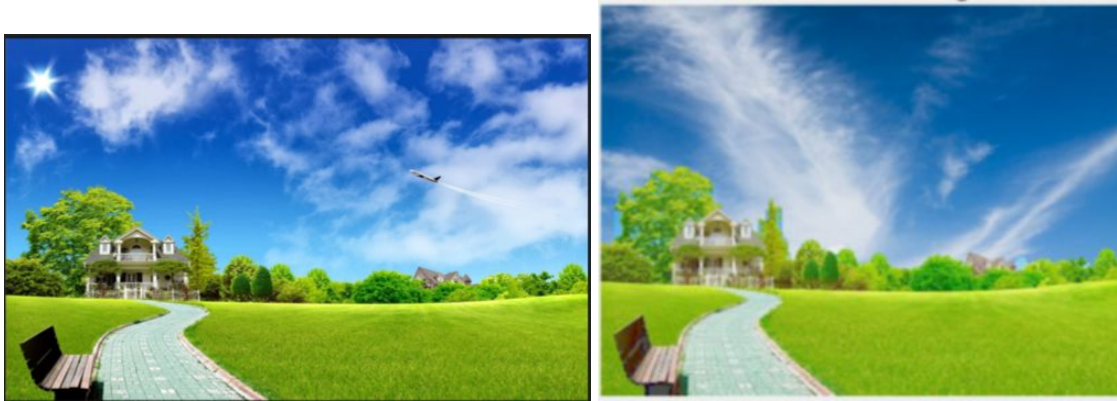
因此我們思索了一下，嘗試了第三個方法。

### Method 3 : HSV with “v” Transfer

我們認為我們可以藉由matlab的內建function先將rgb space轉換到hsv的space，並且對hsv space的“v” channel，也就是圖片的亮度，做

Method 1 提到的Step2, 試圖將source image的亮度分佈趨近於target image。

實作結果(右圖)相較於原圖(左圖), 確實將亮度提升了, 在將天空轉換成藍天上, 相較於前兩個方法, 結果好很多。



但由於只對亮度做調整, 如果遇到轉換成黃昏, 會需要對地景做黃昏的color transformation的狀況, 雖然另一方面, 理論上附帶性的, 地景亮度會降低, 實際實作上, 地景亮度也降低了, 但卻沒能做到將黃昏時的色調投射到地景上的轉換。



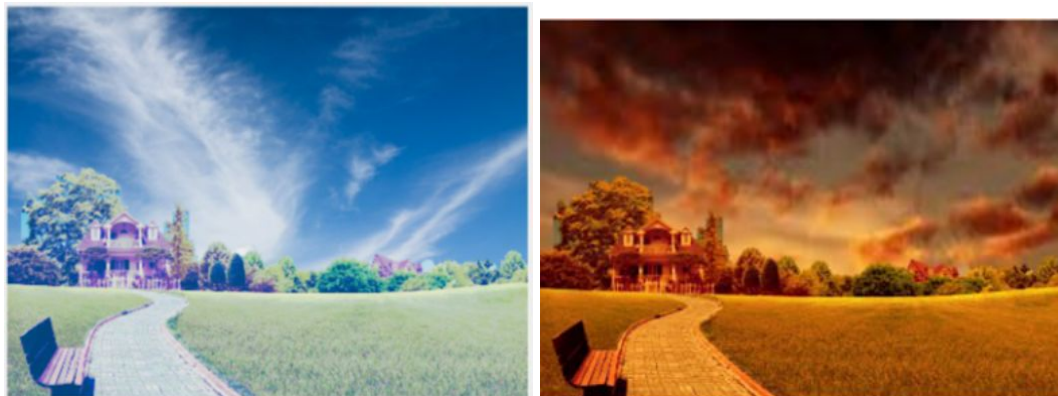
## Method 4 Histogram Matching

這個方法也是屬於color transformation的一種實作方法, 目的是讓source image各個pixel的rgb值的histogram分佈, 趨近於target image中, 各個pixel的rgb值的histogram分佈。

起初我們想說, 針對rgb三個channel各自呼叫matlab提供的histogram equalization的內建function, 使source image中, 各pixel的r, g, b三個channel各自分佈, 趨近於target image各pixel在三個channel的各自分佈。然而後來結果壞掉了, 我們想了一下, 一個pixel的顏色會由三個channel相互影響而成, 然而我們對3個channel各自做histogram equalization, 若3個channel各自拉伸的比例落差很大, 很容易導致結果圖上, 有需多失真的色彩出現。因此我們認為, 應該要對整體圖片做



histogram equalization, 並且我們找到了一個imhistmatch的function, 可以對整張圖做整體性的histogram equalization。



實作結果, 就目前為止, 實作color tranformation的方法中, 呈現結果是感覺上最好, 色調分佈最自然和諧的。

## Method 5 Color Transfer in Correlated Space

最後這個方法是希望可以改進Reinhard, 其目標是藉由下列一系列的旋轉矩陣, 轉置矩陣, 拉伸矩陣相乘, 企圖達到不需要轉成lab space, 直接在rgb space就可以完成將source image顏色數值, 及各點色差分佈, 趨近於target image的目標。

$$I = T_{src} \cdot R_{src} \cdot S_{src} \cdot S_{tgt} \cdot R_{tgt} \cdot T_{tgt} \cdot I_{tgt}$$

$$Cov = U \cdot \Lambda \cdot V^T$$

$$T_{src} = \begin{pmatrix} 1 & 0 & 0 & t_{src}^r \\ 0 & 1 & 0 & t_{src}^g \\ 0 & 0 & 1 & t_{src}^b \\ 0 & 0 & 0 & 1 \end{pmatrix}, T_{tgt} = \begin{pmatrix} 1 & 0 & 0 & t_{tgt}^r \\ 0 & 1 & 0 & t_{tgt}^g \\ 0 & 0 & 1 & t_{tgt}^b \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_{src} = U_{src}, R_{tgt} = U_{tgt}^{-1}$$

$$S_{src} = \begin{pmatrix} s_{src}^r & 0 & 0 & 0 \\ 0 & s_{src}^g & 0 & 0 \\ 0 & 0 & s_{src}^b & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, S_{tgt} = \begin{pmatrix} s_{tgt}^r & 0 & 0 & 0 \\ 0 & s_{tgt}^g & 0 & 0 \\ 0 & 0 & s_{tgt}^b & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where  $t_{src}^r = \bar{R}_{src}$ ,  $t_{src}^g = \bar{G}_{src}$ ,  $t_{src}^b = \bar{B}_{src}$ ,  $s_{src}^r = \lambda_{src}^R$ ,  $s_{src}^g = \lambda_{src}^G$ ,  $s_{src}^b = \lambda_{src}^B$ ,  $t_{tgt}^r = -\bar{R}_{tgt}$ ,  $t_{tgt}^g = -\bar{G}_{tgt}$ ,  $t_{tgt}^b = -\bar{B}_{tgt}$ ,  $s_{tgt}^r = 1/\sqrt{\lambda_{tgt}^R}$ ,  $s_{tgt}^g = 1/\sqrt{\lambda_{tgt}^G}$ , and  $s_{tgt}^b = 1/\sqrt{\lambda_{tgt}^B}$ .

會對source image中的每一個pixel做左上式運算, I 為一矩陣, 當中包含經過一系列矩陣運算後, 求出來的轉換過的rgb值。

右圖及左下圖, 為左上式運算上, 會需要用到的矩陣們的內容。

## Overall Compare

圖中標示的Lab transfer即為Method 1, Reinhard為Method 2, HSV with v change為Method 3, Histmatch為Method 4, CFX為Method 5,

CFX<sub>r</sub> 為CFX的改良版，在做運算時，會多做一個處理，儘可能match target image旋轉矩陣的各column之間的關係to source image旋轉矩陣的各個column。

簡而言之，整體上我們認為imhistmatch(Method 4)在color transformation上，平均表現最好，而只有在換成藍天時，HSV with “v” transfer(Method 3)表現最好。

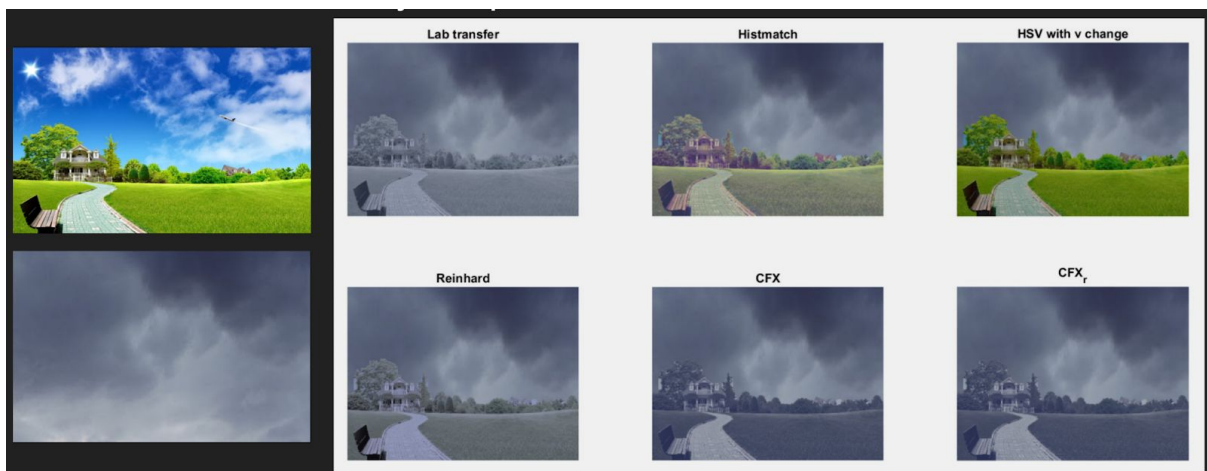
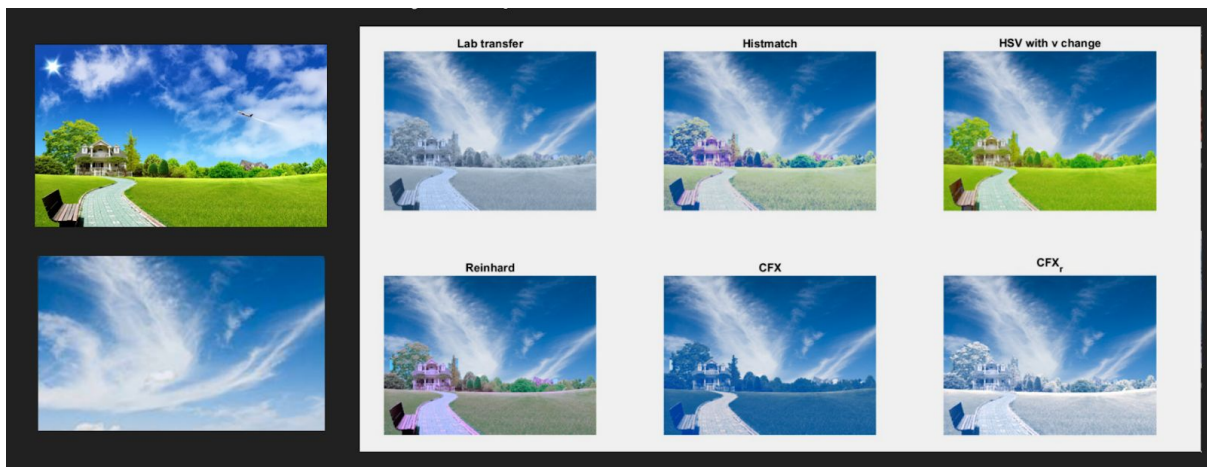
### All color transfer way compare



original view



target sky



- All color transfer way compare





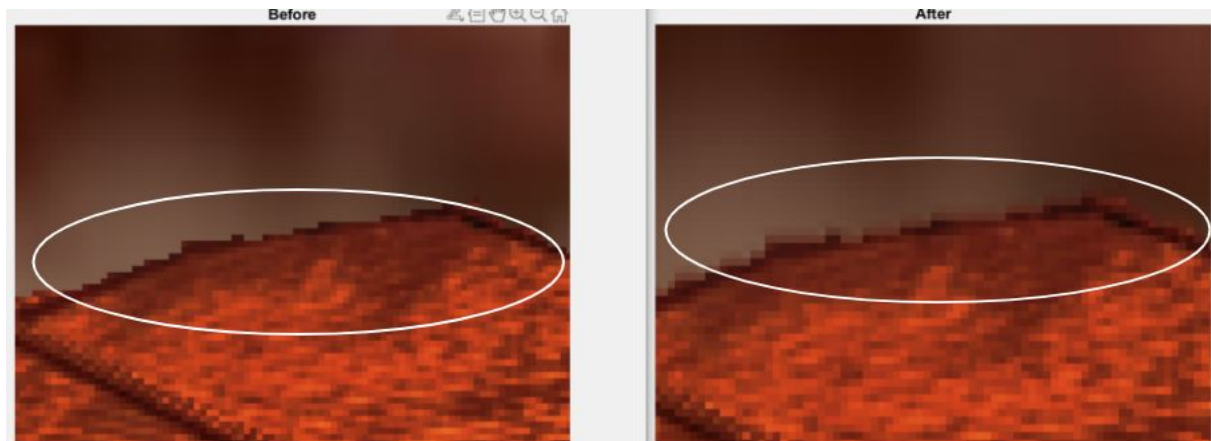
## Step4 : Sky Replacement

依序掃過input圖片的每一個column，將分界線以上的pixel替換為reference sky pixel。

## Step5 : Boundary Smoothing

這裡簡單的使用box filter + Gaussain filter (兩層filter)，達到smoothing邊界的效果。

下圖為結果圖 (兩個都是屋頂的部分)



# Improved After Demo

比較以上這麼多種realistic style transfer，我們可以發現在藍天白雲(晴朗)的時候，只需要更動圖片的明亮度即可，而在另外的情況，再使用histogram matching來做轉換。因此，我們新增了一個自動判斷的功能，當reference的天空是屬於藍天白雲類型的話，則只更動明亮度，反之則使用histogram matching。

主要根據以下兩點來做判斷：

1.variance

2.顏色&明亮度

因為晴朗無雲的天空顏色相對單純，變化性應該會很小，因此我們先將圖片轉至hsv後，計算h的variance，再經過多張圖片的實驗後，我們發現variance  $\leq 0.01$ 有很高機率是藍天白雲系列。

再來是針對顏色組成下去判斷，晴朗無雲的天空大多是由亮白色與亮藍色組成，因此跑一個雙重for迴圈，判斷每個pixel是否是在藍色區間且明亮度超過0.5，這裡也是經過多張圖片實驗後，發現大概要超過80%的pixel才算符合。

因此我們集合以上兩點，多實作一個function在顏色轉換前做判斷，然後根據判斷的結果，選擇不同的演算法下去轉換。

底下為判斷得結果圖



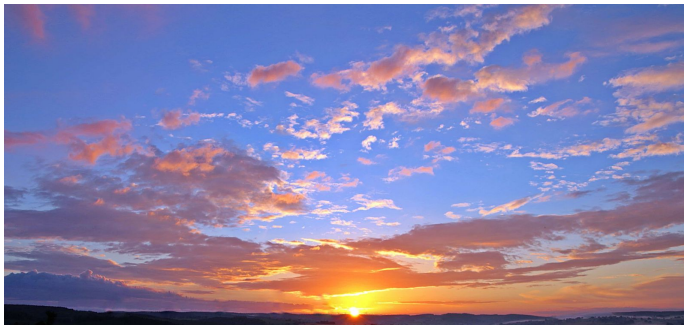
```
>> Auto  
variance: 0.00018702  
pixel proportion: 0.99188  
True
```



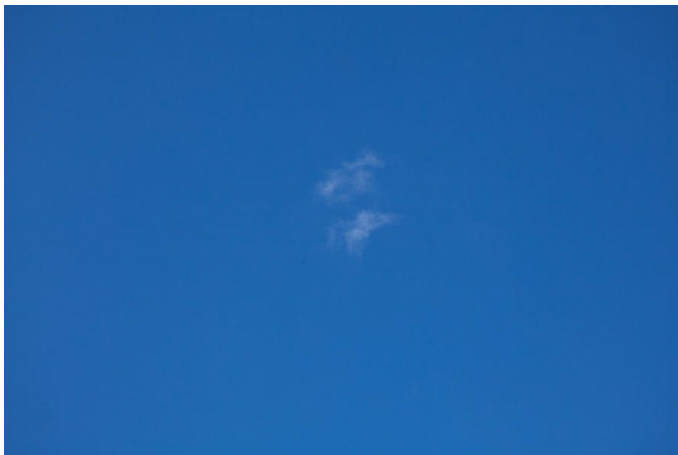
```
>> Auto  
variance: 0.006523  
pixel proportion: 0.47504  
False
```



```
>> Auto  
variance: 0.15204  
pixel proportion: 0  
False
```



```
>> Auto  
variance: 0.065637  
pixel proportion: 0.6026  
False
```



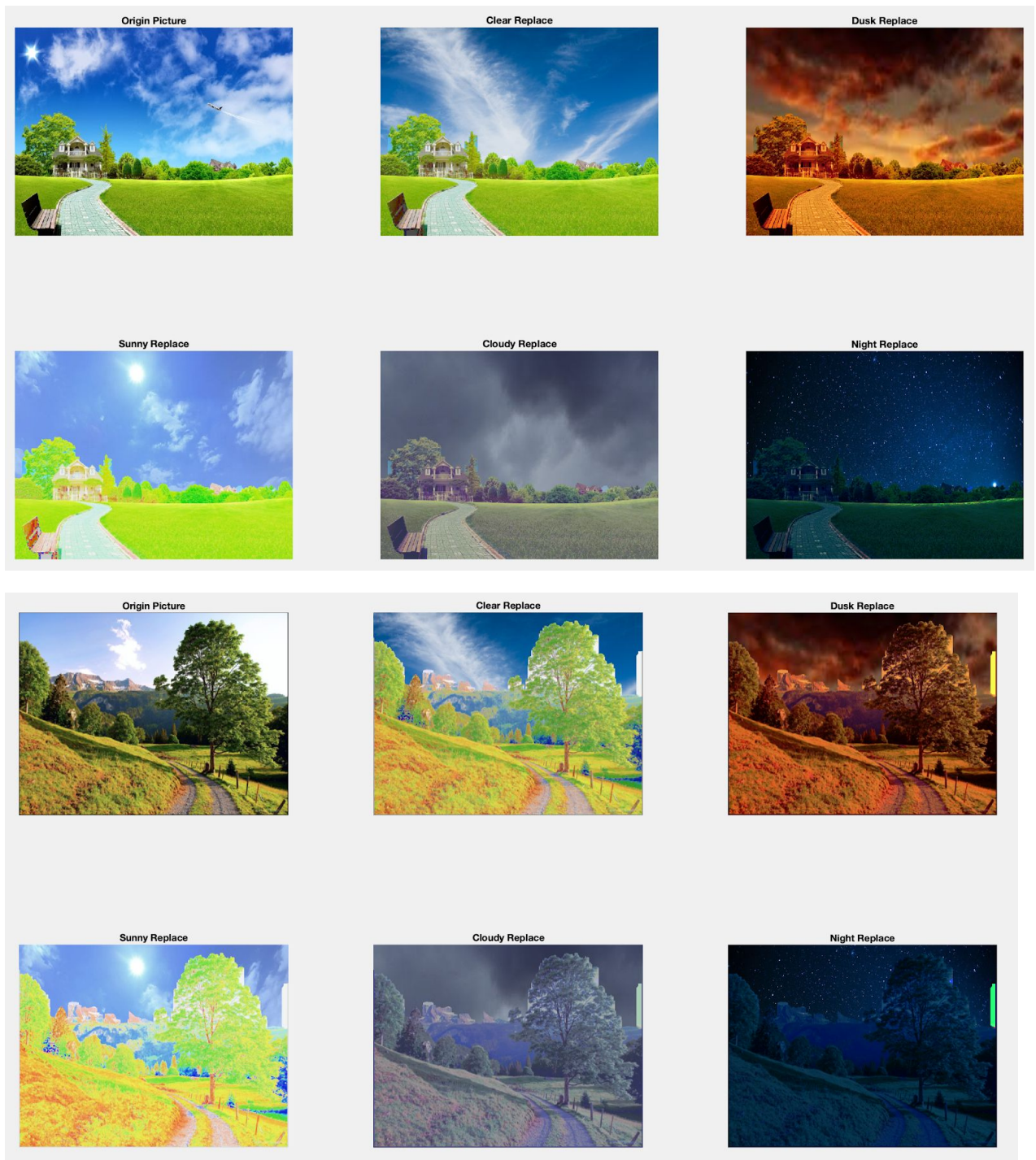
```
>> Auto  
variance: 6.027e-06  
pixel proportion: 1  
True
```



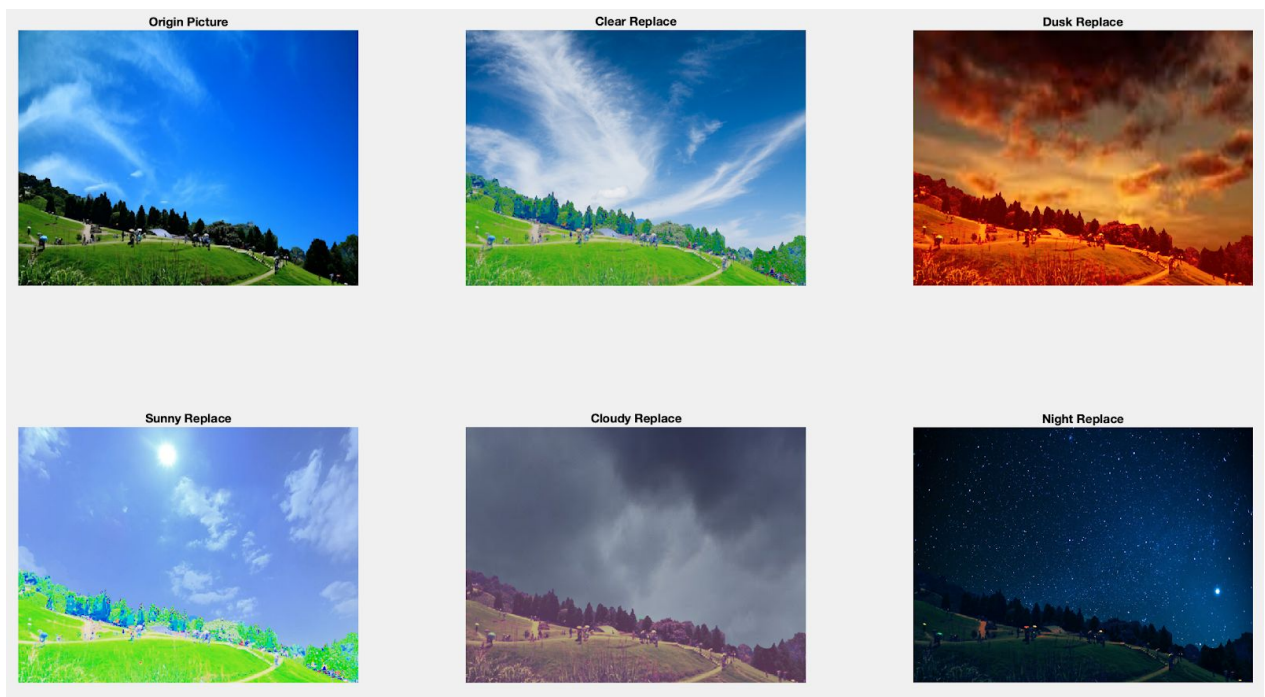
```
>> Auto  
variance: 0.0080497  
pixel proportion: 0.81642  
True
```



# Results & Analysis



分析: 這個轉換結果應該是三個裡面最失敗的, 我們認為可能是因為原圖屬於比較難分天空與地面的界線, 因此在完成sky replacement的流程後, 圖片整體看起來會比較不和諧。



統一分析: 圖片轉換的結果都還不錯，clear sky那張也很完整地保留了原本的色調，只有修改亮度的部分。但是Sunny sky那張，在三張不同的result下，則是顯得有點過曝。

# Conclusion

此次project主要分為兩大重點: Sky segmentation與Realistic Sky Style Transfer。

Sky segmentation: 先利用superpixel降低image pixel的數量 -> 將圖片從rgb轉為hsv space -> 設threshold尋找boundary (將input圖片限制為山景圖的效果會最佳)

Realistic Sky Style Transfer: 如果天空是萬里無雲或是艷陽高照的話，單純修改圖片HSV space中的v值(明亮度)，呈現的效果最為真實，否則的話，使用histogram match的方式，效果會最佳。

# Discussion

1. 嘗試過這麼多種sky style transfer的方法後，發現結果再好的圖片，都會有一定程度的失真。因此，我們認為如果要做出與paper一樣好的結果，勢必要使用其semantic-aware transfer的想法，選擇的reference input要有地面作為轉換的參考，並對於地面的部分先進行簡易的segmentation(可用superpixel或kmeans)，然後再找出input與reference地面間的transfer function，才能達到最真實的效果(即是不能用一種固定的公式來做轉換，必須針對兩張圖片的特性來找出適合的transfer function)。

2. 在Segmentation的部分，其實我們做了很多的假設，才能產出不錯的結果，但如果今天的照片是任意收集的話，那效果就會變得非常差，而且遇到那種天空與地面是分成很多塊的時候，目前的演算法是無法找出最佳解的，因此如果要達到像paper一樣的general的話，可能還是需要依賴AI的方式去計算，但如果是有特定類型的圖片種類時，綜合superpixel,kmeans, boundary detection等上課所學，應該也能做出良好得結果。



# Reference

[http://article.nadiapub.com/IJSIP/vol8\\_no3/26.pdf](http://article.nadiapub.com/IJSIP/vol8_no3/26.pdf)  
[http://vllab.ucmerced.edu/ytsai/SIGGRAPH16/siggraph16\\_sky\\_edit.pdf](http://vllab.ucmerced.edu/ytsai/SIGGRAPH16/siggraph16_sky_edit.pdf)  
[http://www.kalyans.org/research/2016/SkyReplacement\\_SIG16.pdf](http://www.kalyans.org/research/2016/SkyReplacement_SIG16.pdf)  
<https://ieeexplore.ieee.org/document/6703710>  
[https://www.researchgate.net/publication/220805333\\_Color\\_transfer\\_in\\_correlated\\_color\\_space](https://www.researchgate.net/publication/220805333_Color_transfer_in_correlated_color_space)  
[https://www.researchgate.net/publication/221523265\\_Progressive\\_histogram\\_reshaping\\_for\\_creative\\_color\\_transfer\\_and\\_tone\\_reproduction](https://www.researchgate.net/publication/221523265_Progressive_histogram_reshaping_for_creative_color_transfer_and_tone_reproduction)  
<https://journals.sagepub.com/doi/pdf/10.5772/56884>  
<https://github.com/SaumyaRawat/Semantic-Aware-Colour-Transfer/blob/master/Report.pdf>  
<https://www.mathworks.com/matlabcentral/fileexchange/42580-reinhard-stain-normalization>  
<https://www.jianshu.com/p/421f4171d48a>  
[https://blog.csdn.net/sin\\_geek/article/details/45723475](https://blog.csdn.net/sin_geek/article/details/45723475)  
<https://www.cs.tau.ac.il/~turkel/imagepapers/ColorTransfer.pdf>  
<https://www.mathworks.com/help/images/histogram-equalization.html>  
<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6909823>  
<https://wenku.baidu.com/view/4dcb5643336c1eb91a375dbc.html>  
<https://medium.com/codezest/super-fast-color-transfer-algorithm-bd1a76bc7619>  
[https://static.aminer.org/pdf/PDF/000/540/062/rendering\\_natural\\_waters.pdf](https://static.aminer.org/pdf/PDF/000/540/062/rendering_natural_waters.pdf)  
<http://www.yxkxyghx.org/EN/abstract/abstract2352.shtml>