(Previous): Convert the system into the y, cb, cr coordinates



(1) Initialization:

Choose *K* center points randomly.

Denote the center by (*mk*, *nk*) and denote the color by (*yk*, *cbk*, *crk*)

*k* = 1, 2, …, *K*

(2) For each pixel *m*, *n* calculate



*k* = 1, 2, …, *K*

*λ*1 and *λ*2 are adjustable.

In usual, *λ*2 < 1

*λ*1 is also smaller and the importance of the location is determined by *λ*1

(3) Assign the pixel [*m*, *n*] to the *h*th region if

 for all *k* ≠ *h*.

(4) Modify (*mk*, *nk*) and (*yk*, *cbk*, *crk*) for each region

, , ,   
 , 

Repeated (2)-(4) with enough number of times (e.g., 15 times)

進一步的改良

(i) 首先，是initial 的中心點

微調至 gradient 最小的點

先做 x 和 y 方向的 edge detection，得出 gx, gy

再算 g[m, n] = (gx[m, n]^2 +   gy[m, n]^2)^0.5

如果 [mk, nk] 為其中第k 個中心點

則在  [mk + a, nk + b],  a, b = -L, -L+1, ...., L 的範圍內

找尋  g[m, n] 值最小的點

並取代原本的 [mk, nk] 為新的中心點

(ii) 增加 K 值試著將影像分成更多的區域

一般會分為500區左右，K=500

這樣雖然會增加運算量

但我們可以在第二步當中設定

只和 abs(mk - m) < M/K^0.25 且 abs(nk - n) < N/K^0.25 的點相比

(M, N) 是影像的 size

(iii) 調整第二步的 lambda1 和   lambda2 的值

讓效果可以更好

(iv) 最後的結果，若沒有 contact 則再做細分

例如，如果第k 區其實是二個以上的不相鄰區域

則將第k 區再做細分

(先二值化，再用和上次 fast scanning 的方法做分割

類似於Matlab 當中 bwlabel 的指令)

若使用 Python, binary segmentation 的指令為

from skimage import measure

B = measure.label(A)