**以DTW 搜尋最佳特徵點演算法**

**Function:**

 Inputs:

1. *Gray*: 經過多點校正後的灰階影像
2. *InitXYs*: 初始地標點位置(上一次iteration重建後的地標點位置，若為第一次iteration則以平均臉作為初始位置)
3. *NoOfCandidates*:沿法線方向搜尋的候選點個數
4. *BlockSize*: 特徵區塊大小(*BlockSize* x *BlockSize*)
5. *FeatMean*: 各地標點訓練後的平均特徵向量，(*FeatMean*[L]代表第L個地標點的平均特徵向量，其維度為1xBlockSize2

**

1. *FeatStandardDev*: 各地標點區塊特徵訓練後各個維度之標準差

)  
其中M為樣本數(61)

**Outputs:**

1. *XYs*: 未使用Dynamic Programming找到的最佳特徵點位置
2. *BestXYs:* 使用Dynamic Programming搜尋到的最佳特徵點位置
3. *Errs*: 各個最佳特徵點的比對誤差

**Algorithm:**

/\* 初始化DTW所要用的資料 \*/

FOR m = 0 to *NoOfLandmarks-1*

Let P = (*x*,*y*) = (*InitXYs*[2\*m-1],*InitXYs*[2\*m]); /\*第m個地標點位置\*/

Let PreviousP = (*Px*, *Py*); /\*地標點P的前一個地標點位置\*/

Let NextP = (*Nx*,*Ny*); /\*地標點P的下一個地標點位置\*/

Let ; /\*法線方向單位向量\*/



FOR k = -*p* to *p*





Extract the block feature at point (*CX*[*m,k+p*]*,CY*[*m,k+p*]) as ;

; /\*如果(CX,CY)的特徵區塊超出影像邊界，則NodeErrs[m][k+p]=999999,0 \*/

END

CompMarkList[0] = {0, 1, 2, 3, …, 12}; /\*下巴的地標點編號序列\*/

CompMarkList[1] = {13,…,20}; /\*右眼的地標點編號序列\*/

CompMarkList[2] = {}; /\*左眼的地標點編號序列\*/

CompMarkList[3] = {0, 1, 2, 3, …, 12}; /\*右眉的地標點編號序列\*/

CompMarkList[4] = {13,…,20}; /\*左眉的地標點編號序列\*/

CompMarkList[5] = {};/\*嘴巴的地標點編號序列\*/

CompMarkList[6] = {};/\*鼻子的地標點編號序列\*/

Copy NodeErrs to AccErrs;

FOR c = 0 to 6 /\* 五官逐一個別獨立執行DTW \*/

MarkList = CompMarkList[c];

last = length(MarkList) - 1; /\* 五官地標點個數 \*/

*R* = 4; /\* 定義DTW每個節點(n,p)之前繼來源節點為(n-1,p-R), …,(n-1,p-1), (n-1,p), (n-1,p+1), …,(n-1,p+R) \*/

FOR k=MarkList[1] to MarkList[last] /\* 開始DTW 演算法 \*/

FOR pp = 0 to 2\*p

IF (pp<=R-1) THEN /\* 部分前繼來源(k, pp-j)會有pp-j<0的狀況 \*/



 /\*紀錄最佳來源 \*/

ELSE IF (pp>=2\*p+1-r) /\* 部分前繼來源(k, pp+j)會有pp+j>2p的狀況 \*/



/\*紀錄最佳來源 \*/

ELSE



/\*紀錄最佳來源 \*/

ENDIF

END FOR

END FOR

 /\*開始從最佳終點依序由最佳來源回溯出最佳路徑\*/

BestXYs(2\*NoOfLandmarks-1) = CX(MarkList(last), *Ind*);

BestXYs(2\*NoOfLandmarks) = CY(MarkList(last), I*nd*);

Errs(MarkList(last)) = NodeErrs(MarkList(last), *Ind*);

FOR k = MarkList(last):-1:MarkList(2)

*Ind* = *BestInd*(k, *Ind*);

Errs(k-1) = NodeErrs(k-1, *Ind*);

BestXYs(2\*(k-1)-1) = CX(k-1, *Ind*);

BestXYs(2\*(k-1)) = CY(k-1, *Ind*);

END FOR

END FOR