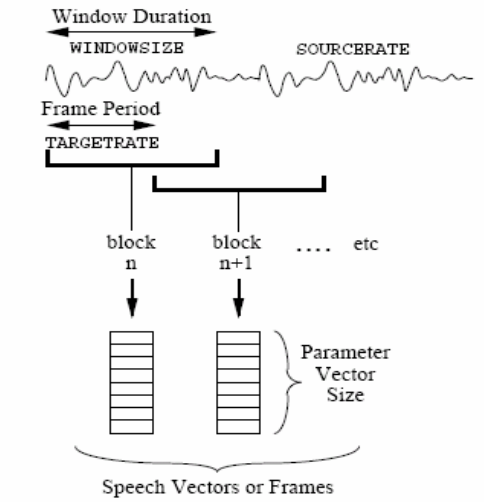
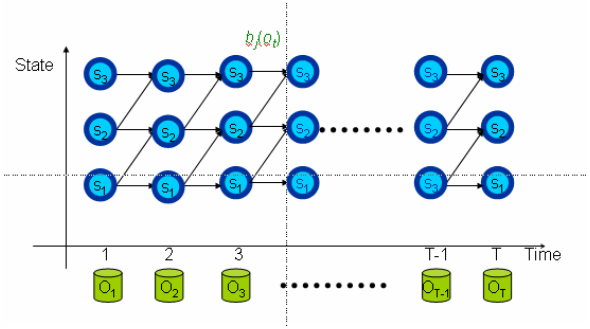
```
//////////////////////////////////////////
int   Len[MAX_NAME]; //存放每個人名的HMM的model數
int HMM_ID_List[MAX_NAME][MAX_LEN]; //存放每個人名所用到的HMM的model的編號ID
float Delta[MAX_NAME][MAX_LEN][MAX_STATE]; //Viterbi 中的 delta(t)
float Score[MAX_NAME]; //每個人的分數
char  Name[MAX_NAME][20]; //儲存人名的中文字

int   Name_No; //實際人名的個數
//////////////////////////
 ……….

  getcwd(Directory,200);  //取得本程式執行位置
   ///////初始工作 ////////////////
   Allocate_Model_Memory((int)151); //Allocate HMM Model Memory,初設最多有 160 HMM
   Set_Const();//for feature extraction
  ////////讀入 HMM models//////////////////////
   sprintf(ModelPath,"%s\\HMM_model",Directory);
                  printf("%s \n",ModelPath);
   Load_HMM_FILES(ModelPath);
   printf("Model[total=%3d] Loading OK...\n",Total_HMM);
   ////////讀入 每個人名及其對應之 HMMs ////////////////////
        Read_Name_Information();
   /////////讀入 wave 檔 辨識//////////
   if ((WAVSRC = fopen(buf,"rb"))==NULL)
      {
           printf("Open WaveFile: %s Error!!\n",buf);
           getchar();
           exit(1);
      }
      filesize=filelength(fileno(WAVSRC));
      speechdata=(short *)data;
      fread(data,filesize,1,WAVSRC);
      fclose(WAVSRC);
      Front_End=0;
      Back_End=filesize/2; //因爲每一個 sample爲2 bytes
```

```
struct Model_struct
 {
     char   Name[20];
     short  State;
     short  Phone_ID;
     float  Mean[Num_State][Max_Mix][Vec_Size];
     float  Var[Num_State][Max_Mix][Vec_Size];
     float  Trans[Num_State][Num_State];
     float  GConst[Num_State][Max_Mix];
     float  Mix_Weight[Num_State][Max_Mix];
     short  Mix[Num_State];

 } ;
 struct Model_struct *Model;
 //存HMM models的 structure
```

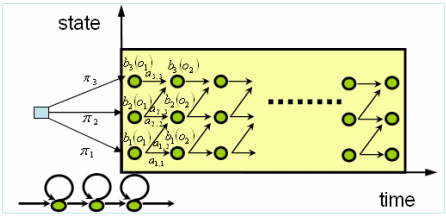
```
Total_Frame=Feature(); //Caculate feature vector
                       //feature vector存在  Cep_Data1[j][i]
                       //j是time index, i 是 vector dimension index
 printf("Total Frame Num (Total feature vector Num)=%d\n",Total_Frame);

 ////////////////////////////////
  for (i=0;i<Total_Frame;i+=1) //Caculate model state observation in advance
    for (j=0;j<Total_HMM;j++)
        for (k=1;k<Model[j].State-1;k++)
              Get_Fil_B(i,j,k);
  printf("State Observation Calculation OK\n");
 //結果以log(prob)（取log）形式存在存在  B_O[Max_Frame_Num][160][Num_State];
/////////////////Initializatiog for Viterbi Search ////////////////////
```




```
 for(name_id=0;name_id<Name_No;name_id++)//Names
   for(int model_len=0;model_len<Len[name_id];model_len++)//HMM models for a specific name
    for(int k=0;k<Model[HMM_ID_List[name_id][model_len]].State;k++) //HMM state
          Delta[name_id][model_len][k]=(float)Min_Delta;

 for(name_id=0;name_id<Name_No;name_id++)
                Delta[name_id][0][0]=(float)0.0;
```



```
for(frame=0;frame<Total_Frame;frame++) //輸入語句有多少個feature vectors
 {
   for(name_id=0;name_id<Name_No;name_id++) //對於所有人名展開HMM平面
   {
     for(model_len=Len[name_id]-1;model_len>=0;model_len--) //HMM models for a specific name
    {
       for(k=Model[HMM_ID_List[name_id][model_len]].State-2;k>=1;k--)//HMM state
       {
         val1=Delta[name_id][model_len][k-1]+Model[HMM_ID_List[name_id][model_len]].Trans[k-1][k];//來自前一個時間、前一個狀態
         val2=Delta[name_id][model_len][k]+Model[HMM_ID_List[name_id][model_len]].Trans[k][k];//來自前一個時間、同一個狀態
         if(val1 > val2)
             max_val=val1;
          else
             max_val=val2;
          Delta[name_id][model_len][k]=max_val+B_O[frame][HMM_ID_List[name_id][model_len]][k];
       } //for HMM state

       Delta[name_id][model_len][0]=Delta[name_id][model_len-1][Model[HMM_ID_List[name_id][model_len-1]].State-2]
          +Model[HMM_ID_List[name_id][model_len-1]].Trans[Model[HMM_ID_List[name_id][model_len-1]].State-2]
                                                          [Model[HMM_ID_List[name_id][model_len-1]].State-1];
     } //for Model_len
     Delta[name_id][0][0]=(float) Min_Delta;
     Score[name_id]=Delta[name_id][Len[name_id]-1][Model[HMM_ID_List[name_id][Len[name_id]-1]].State-2];
   } //for name_id
 } //for speech frame

 float max_val=(float) Min_Delta;
 Top1_Index=-1;
  for (name_id=0;name_id<Name_No;name_id++)
     {
       if(Score[name_id]>max_val)
       {
          max_val=Score[name_id];
         Top1_Index=name_id;
       }
   }
```



$$f\left(\boldsymbol{X}=\boldsymbol{x}\middle|\boldsymbol{\mu},\boldsymbol{\Sigma}\right)=N\left(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}\right)=\frac{1}{\left(2\pi\right)^{L/2}\left|\boldsymbol{\Sigma}\right|^{1/2}}\exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^{t}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)\right)$$

if a diagonal covariance metrix is assumed

$$N\left(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}\right)\approx\frac{1}{\left(2\pi\right)^{L/2}\begin{vmatrix}\sigma_{1,1} & & & \\ & \sigma_{2,2} & & \\ & & \ddots & \\ & & & \sigma_{L,L}\end{vmatrix}^{1/2}}\exp\left(-\frac{1}{2}\sum_{l=1}^{L}\frac{1}{\sigma_{l,l}}\left(x_l-\mu_l\right)^2\right)$$

$$\approx\frac{1}{\left(2\pi\right)^{L/2}\left(\sigma_{1,1}\times\sigma_{2,2}\times\ldots\times\sigma_{L,L}\right)^{1/2}}\exp\left(-\frac{1}{2}\sum_{l=1}^{L}\frac{1}{\sigma_{l,l}}\left(x_l-\mu_l\right)^2\right)$$

$\therefore$

$$\log N\left(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}\right)\approx-(0.5)\times Gconst\ +-\frac{1}{2}\sum_{l=1}^{L}\frac{1}{\sigma_{l,l}}\left(x_l-\mu_l\right)^2$$

```
void Get_Fil_B(int i, int j, int k)
{
 double   diff[Max_Mix];
 double  MAX;
 int     m, n;
 double  value,value2;
 double temp;

 MAX = 1.0e30;
 double sum=(double)0.0;

 for (m=0;m<Model[j].Mix[k];m++)
 {
  diff[m] =(double) 0.0;
   for (n=0;n<Vec_Size;n++)
   {
    value=(double)Cep_Data1[i][n]-(double)Model[j].Mean[k][m][n];
    diff[m] += value*value*(double)Model[j].Var[k][m][n];
   }
      if(_finite(diff[m])==0||_isnan(diff[m])!=0||diff[m]<(double)1.0E-30)
                    diff[m]=(double)1.0E-30; //check if underflow
        temp=(double)(diff[m]+(double)Model[j].GConst[k][m])*(double)(-0.5);
        value2=exp(temp);
        sum+=value2*(double)Model[j].Mix_Weight[k][m];

 }

 B_O[i][j][k] = (float)log(sum); //stored in log domain

}
```