function [Dist,D,k,w]=dtw(t,r)
% Dynamic Time Warping Algorithm
% Dist is unnormalized distance between t and r
% D is the accumulated distance matrix
% k is the normalizing factor
% w is the optimal path
% t is the vector you are testing against
% r is the vector you are testing

D1=imread('001','tif');D2=imread('002','tif');D3=imread('003','tif');D4=imread('004','tif');D5=imread('005','tif');D6=imread('006','tif');D7=imread('007','tif');D8=imread('008','tif');D9=imread('009','tif');D10=imread('010','tif');D11=imread('011','tif');D12=imread('012','tif');D13=imread('013','tif');D14=imread('014','tif');D15=imread('015','tif');D16=imread('016','tif');D17=imread('017','tif');D18=imread('018','tif');D19=imread('019','tif');D20=imread('020','tif');D21=imread('021','tif');D22=imread('022','tif');D23=imread('023','tif');D24=imread('024','tif');D25=imread('025','tif');D26=imread('026','tif');D27=imread('027','tif');D28=imread('028','tif');D29=imread('029','tif');D30=imread('030','tif');D31=imread('031','tif');D32=imread('032','tif');D33=imread('033','tif');D34=imread('034','tif');D35=imread('035','tif');D36=imread('036','tif');D37=imread('037','tif');D38=imread('038','tif');D39=imread('039','tif');D40=imread('040','tif');D41=imread('041','tif');D42=imread('042','tif');D43=imread('043','tif');D44=imread('044','tif');D45=imread('045','tif');D46=imread('046','tif');
% this is reading the images as matrix, for the MOD09A1 dataset, a total of 46 images was readed,
% assume that the images are n*m
for i=1:n
   for j=1:m
      t=[D1(i,j),D2(i,j),D3(i,j),D4(i,j),D5(i,j),D6(i,j),D7(i,j),D8(i,j),D9(i,j),D10(i,j),D11(i,j),D12(i,j),D13(i,j),D14(i,j),D15(i,j),D16(i,j),D17(i,j),D18(i,j),D19(i,j),D20(i,j),D21(i,j),D22(i,j),D23(i,j),D24(i,j),D25(i,j),D26(i,j),D27(i,j),D28(i,j),D29(i,j),D30(i,j),D31(i,j),D32(i,j),D33(i,j),D34(i,j),D35(i,j),D36(i,j),D37(i,j),D38(i,j),D39(i,j),D40(i,j),D41(i,j),D42(i,j),D43(i,j),D44(i,j),D45(i,j),D46(i,j)];
      r=[0.151144,0.0475944,0.175165,0.157775,0.164598,0.172529,0.155204,0.142113,0.136211,0.145350,0.140974,0.133333,0.182927,0.182927,0.242059,0.204676,0.238500,0.260517,0.276132,0.331067,0.406653,0.440842,0.415762,0.609840,0.694531,0.326094,0.334303,0.409154,0.439298,0.402376,0.385945,0.341566,0.338080,0.308387,0.282548,0.265436,0.257268,0.233965,0.190283,0.0517611,0.197170,0.040494];
   end
end
% this is the vector you are testing against
% r is the vector you are testing
D=abs(repmat(t(:),1,M)-repmat(r(:)',N,1))/10; % this replaces the nested for loops from above

Thanks Georg Schmitz
%d is abs DTW value divided 10 for the convenience of results outputs

D=zeros(size(d));
D(1,1)=d(1,1);

for n=2:N
    D(n,1)=d(n,1)+D(n-1,1);
end

for m=2:M
    D(1,m)=d(1,m)+D(1,m-1);
end

for n=2:N
    for m=2:M
        D(n,m)=d(n,m)+min([D(n-1,m),D(n-1,m-1),D(n,m-1)]);
    end
end

Dist=D(N,M);
n=N;
m=M;
k=1;
w=[];
w(1,:)=[N,M];
while ((n+m)~=2)
    if (n-1)==0
        m=m-1;
    elseif (m-1)==0
        n=n-1;
    else
        [values,number]=min([D(n-1,m),D(n,m-1),D(n-1,m-1)]);
        switch number
            case 1
                n=n-1;
            case 2
                m=m-1;
            case 3
                n=n-1;
                m=m-1;
        end
    end
    k=k+1;
    w=cat(1,w,[n,m]);
end
new(i,j)=Dist;
end
A=im2uint16(new); % the results image DN value are hexadecimal, and the results should divide by 6553.6 to get the final results of DTW distance
imwrite(A,'DTWDIST.tif','tif'); % produce DTW distance image, as shown in Figure 6