

Frechet Distance Demo.....	1
Choose the type of polylines.....	1
Request a leash length.....	2
Compute the Frechet distance and annotate the plot .....	4

## Frechet Distance Demo

```
%-- Script to run the Decision Problem (DP) for Frechet distance
%-- The DP is based on Algorithm 1 in "Computing the Frechet Distance
%-- Between Two Polygonal Curves", Alt and Godau, International Journal of
%-- Computational Geometry and Applications, 1995.
%
%-- Author:      Rich Kenefic
%-- Created:     14-Dec-2011
%-- Modified:    18-Oct-2012 To include the Frechet computation
%
clear all
close all
global I J lP lQ lPQ bP bQ
choice = menu('option','Std.','New Random','Re-Run Last Random');
```

## Choose the type of polylines

```
switch choice
    %--Standard instance
    case 1
        %--default initial state
        S1 = RandStream('swb2712','seed',0);
        %--unused
        S2 = RandStream('swb2712','seed',1);
        %--create a new instance
    case 2
        clk = sum(100*clock);
        S1 = RandStream('swb2712','seed',clk); %--save state
        S2 = RandStream('swb2712','seed',clk+1);
        save FRECHET_starting_clk.mat clk
        fprintf('*****initialization of rand*****\n');
        fprintf('    clk = %d\n',clk);
        fprintf('*****\n\n');
        %--rerun the last instance
    case 3
        load FRECHET_starting_clk.mat clk
        S1 = RandStream('swb2712','seed',clk); %--use last state
        S2 = RandStream('swb2712','seed',clk+1);
end
RandStream.setDefaultStream(S1);
```

```

%-- Create two Polygonal curves, P (man's path) and Q (dog's path)
%-- where P(1,i) and P(2,i) are the (x,y) coordinates of the
%-- ith vertex in P, Q(1,j) and Q(2,j) are the (x,y) coordinates
%-- of the jth vertex in Q

I = randi([5 10]); %--number of vertices in P
J = randi([5 10]); %--number of vertices in Q
for i=1:I
    P(1,i) = randi([0 50]); P(2,i) = randi([0 50]);
end
%--keep the start and stop for P and Q close
Q(1,1) = P(1,1) + randi([-2 2]); Q(2,1) = P(2,1) + randi([-2 2]);
for j=2:J-1
    Q(1,j) = randi([0 50]); Q(2,j) = randi([0 50]);
end
Q(1,J) = P(1,I) + randi([-2 2]); Q(2,J) = P(2,I) + randi([-2 2]);

figure(1)
plot(P(1,:),P(2,:), 'g-', Q(1,:),Q(2,:), 'r:'); hold on;
plot([P(1,1) P(1,1)], [P(2,1) P(2,1)], 'go');
plot([P(1,I) P(1,I)], [P(2,I) P(2,I)], 'gx');
plot([Q(1,1) Q(1,1)], [Q(2,1) Q(2,1)], 'ro');
plot([Q(1,J) Q(1,J)], [Q(2,J) Q(2,J)], 'rx');

plotFSD = 1;
drawnow();

```

## Request a leash length

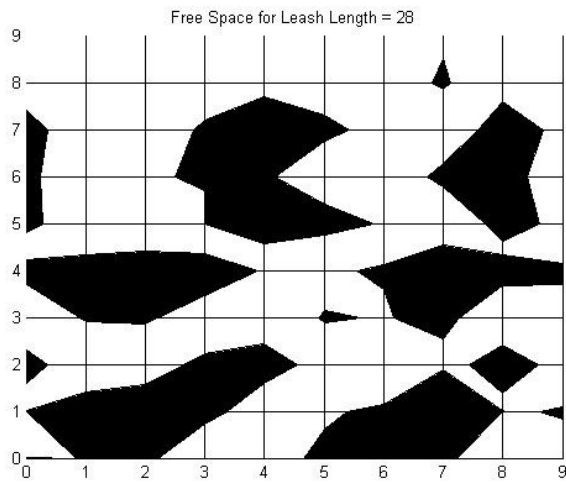
```

dialogstr = 'Enter the leash length';
LEN = inputdlg(dialogstr);
len = str2num(LEN{1});
while ~isempty(len)
    %--decide if the leash is long enough
    decision = frechet_decide(P,Q,len,plotFSD,0);
    drawnow();
    pause(2);
    if decision==0
        rstr = 'The leash is too short';
        fprintf([rstr '\n']);
    else
        rstr = 'The leash is long enough';
        fprintf([rstr '\n']);
    end
    lle = len;
    dialogstr = [rstr ', enter a new guess or just click OK to stop'];
    LEN = inputdlg(dialogstr);
    len = str2num(LEN{1});
end

```

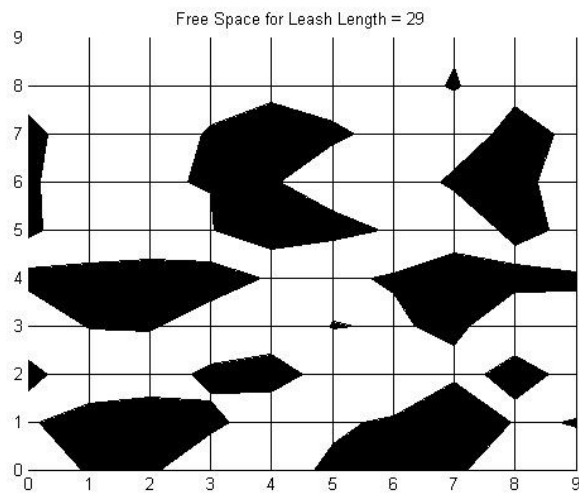
Enter the leash length

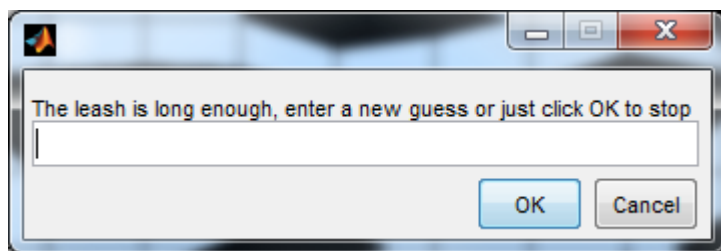
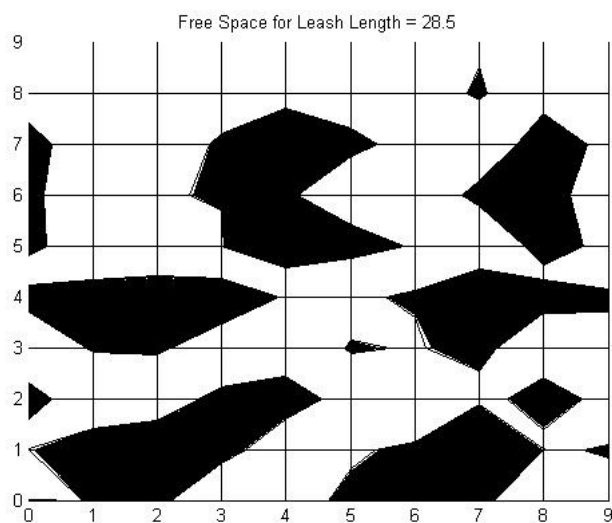
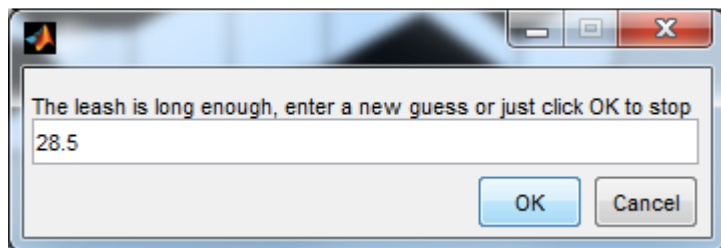
OK Cancel



The leash is too short, enter a new guess or just click OK to stop

OK Cancel





## Compute the Frechet distance and annotate the plot

```
figure(1); legend({'man' 'dog' 'start' 'stop' 'start' 'stop'});
frechet_init(P,Q);
flen = frechet_compute(P,Q,1);
figure(1);
title(['Computed length = ' num2str(flen) ', last guess = ' num2str(11e)]);
```

frechet distance not found in the 2nd pass: len = 28.160256

