%Written by: Frank Lam

%Indiana University - Purdue University Indianapolis

%

%Alexander G. Obukhov,PhD Labratory

%Indiana University School of Medicine

%Cellular & Integrative Physiology

%Select video files and let user label the files

disp("Select Video File.");

[File,sPath] = uigetfile({'\*.mp4';'\*.avi';'\*.mj2';'\*.mpg';'\*.wmv';'\*.asf';'\*.asx';'\*.m4v';'\*.mov';'\*.m4v';'\*.ogg'});

Var = inputdlg("Enter a label for this video");

Var = char(Var);

keepGoingTCheck = 1;

%Script ends if usesr cancels selection one of the files

if (isequal(File,0))

msgbox('You canceled selecting one of the video files. Program has ended.');

else

%Check for number only input

while (keepGoingTCheck == 1)

prompt = ('Enter time interval you want to calculate in seconds (numbers only): ');

t = inputdlg(prompt);

t = str2double(t);

if (isnan(t) == 0)

keepGoingTCheck = 0;

end

end

%Track points and gather data

[dist,rate,dIntC] = trackPoints(File,t,Var);

rIntC = dIntC;

rIntC(1) = [];

%Plotting figures

figure('Position',[100,100,900,700]);

subplot(121); scatter(rIntC, rate);

title(sprintf('Rate vs. Time (%s)', Var));

xlabel("Time in Video (seconds)");

ylabel('Rate(\Delta euclidean distance between frames/\Delta time)');

subplot(122); scatter(dIntC, dist);

title(sprintf('Distance vs. Time (%s)', Var));

xlabel("Time in Video (seconds)");

ylabel('Euclidean Distance');

%Saving data to a excel file

filename = sprintf('%s\_Data.xlsx', Var);

mkdir PointTrackerData;

rCompiled = [rIntC;rate];

dCompiled = [dIntC;dist];

excelFileName = filename;

fullFileName = fullfile('PointTrackerData', excelFileName);

xlswrite(fullFileName,{'Seconds','Rate'},'Rate','A1');

xlswrite(fullFileName ,transpose(rCompiled),'Rate','A2');

xlswrite(fullFileName ,{'Seconds','Distance'},'Distance','A1');

xlswrite(fullFileName ,transpose(dCompiled),"Distance",'A2');

end

%Track points and gather data function

function [dist,rate,timeIntC]= trackPoints(videoFile, timeInterval,type)

video = vision.VideoFileReader(videoFile);

vidAtt = VideoReader(videoFile);

videoPlayer = vision.VideoPlayer('Position',[100,100,700,500]);

%choose object bound parameter in video frame

objectFrame = video();

%Choose points on first video frame to track

figure('Position',[100,100,700,500]);

imshow(objectFrame);

title1 = sprintf('Select interest points on %s video', type);

title(title1);

hold on;

pointsCheck = 1;

while pointsCheck == 1

[x,y] = ginput(2);

close();

selPoints = [x(:),y(:)];

cPoints = cornerPoints(selPoints);

%display selected points and prompt user to confirm the points

pointImage = insertMarker(objectFrame,cPoints.Location,'o','Color','green');

figure('Position',[100,100,700,500]);

imshow(pointImage);

title2 = sprintf('Interest points selected on %s video', type);

title(title2);

answer = questdlg("Would you like to start with the two points shown?",'Confirmation','Start','Reselect Points','Start');

if strcmp('Start',answer)

pointsCheck = 0;

close();

%user selects "Reselect Points"

else

close();

pointsCheck = 1;

imshow(objectFrame);

title1 = sprintf('Reselect interest points on %s video', type);

title(title1);

hold on;

end

end

%create tracker object using Kanade-Lucas-Tomasi algorithm and initialize it

tracker = vision.PointTracker('NumPyramidLevels',4,'MaxIterations',50);

initialize(tracker,cPoints.Location,objectFrame);

cArray = [];

p1x = [];

p1y = [];

p2x = [];

p2y = [];

%add initial frame points

p1x(end+1)= selPoints(1,1);

p1y(end+1)= selPoints(1,2);

p2x(end+1)= selPoints(2,1);

p2y(end+1)= selPoints(2,2);

cArray(end+1) = 0;

counter = 0; %count number of times data points were added

fCounter = 0;

frameRate = round(vidAtt.FrameRate);

%Read,track,display points, and the results in each video frame

while ~isDone(video)

frame = video();

fCounter = fCounter + 1;

[points,validity] = tracker(frame);

%Get frame points data at specific frame intervel in terms of seconds

if(fCounter == (frameRate\*timeInterval))

counter = counter + 1;

cArray(end+1) = counter;

p1x(end+1)= points(1,1);

p1y(end+1)= points(1,2);

p2x(end+1)= points(2,1);

p2y(end+1)= points(2,2);

fCounter = 1;

end

out = insertMarker(frame,points(validity, :),'o');

videoPlayer(out);

end

distArray = [];

cDArray = [];

%calculating euclidean distance

for x =1:(counter+1)

eDist = sqrt((((p2x(x)-p1x(x)).^2) + (((p2y(x)-p1y(x))).^2)));

distArray(end+1) = eDist;

end

%calcuating rate; the change of distance (between the two points) in

%specified intervals of time

for c =1:(length(distArray)-1)

cDArray(end+1) = abs(distArray(c+1)- distArray(c));

end

rateArray = cDArray./(timeInterval);

%Function output

rate = rateArray;

cArray = cArray.\*timeInterval; %convert to seconds

timeIntC = cArray;

dist = distArray;

release(videoPlayer);

release(video);

end