ARRAY DETECTION AND TRACKING

Submitted by
Aldrin Wilfred Arokiasamy
[redacted]

Description

To share the particular vehicles information to the following vehicles, vehicle to vehicle communication is needed. The primary vehicles data were collected from its sensors and transmitted to the subordinate vehicles. For communicating huge data between vehicles, we need Multiple Input Multiple Output Radio Frequency (MIMO-RF) transceiver that is complex and has increased design cost. The drawback can be overcome with the use of Visual Light Communication (VLC), for the reason it is easily accessible and has high probability of seamless communication. In this simulation, a Light Emitting Array (LEA) arrangement was placed. The LEA acts as the transmitter, which was the required Region of Interest (ROI). The ROI was located using the Harris edge-corner detection method. The cameras placed in the secondary vehicles were the receivers in this case, where the ROI has to be extracted. To reduce the transmission error rate at the receiver processing unit, the Kalman filter tracking was used to locate and track the LEA. In this MATLAB simulation, for our convenience, we assume that the LEA moves horizontally or vertically, and the idea of Kalman Filter methodology was proposed along with geometry techniques to improve the locating and tracking of the LEA. The expected simulation results will show the tracking plots for both the schemes of Kalman filter with and without geometric properties.

LEA arrangement:

In this simulation model, the LEA plays a main role for communication. The LEA arrangement has the array of LEDs that switches ON and OFF according to the output data from various sensors of the primary vehicle. For our convenience, the LEA arrangement has black outer boundary and white inner boundary as shown in figure 2.

LEA locating using Harris edge-corner detection:

The Harris corner detection method was used to obtain the sharp intensity changes, thus locating the corners of the LEA. In this simulation model, the LEA

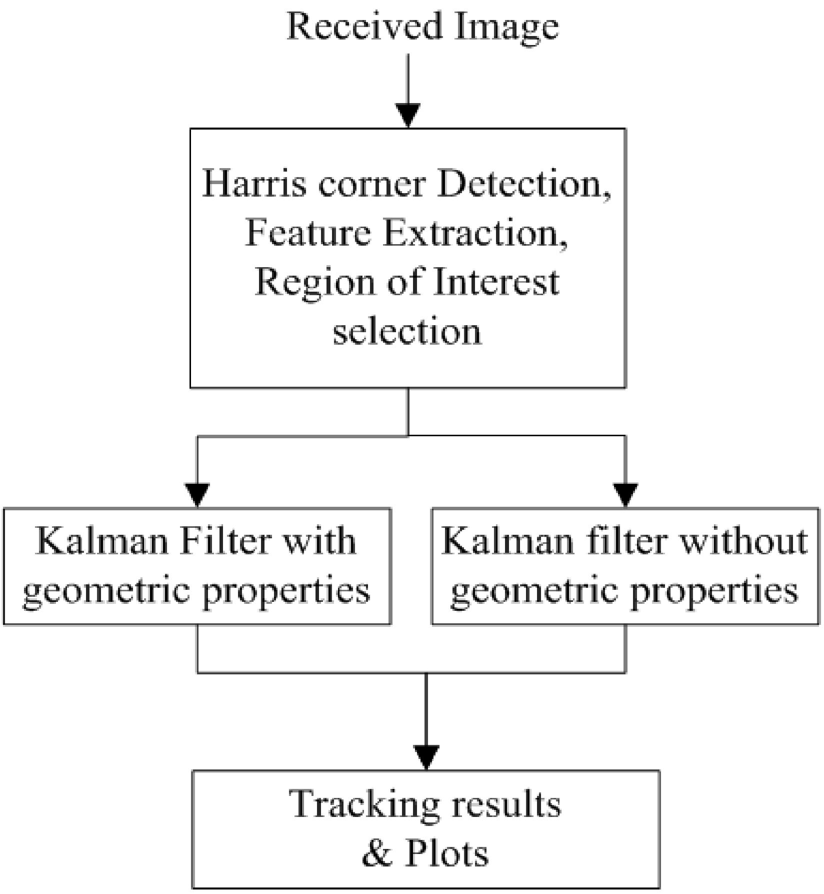


Figure 1: Implemented Algorithm.

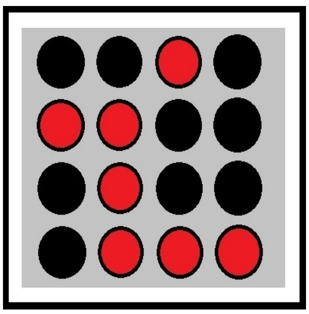


Figure 2: LEA arrangement with black and white boundary.

Figure 3: Background utilized for LEA detection and tracking method.

Figure 4: LEA with black and white boundary in a complex background.

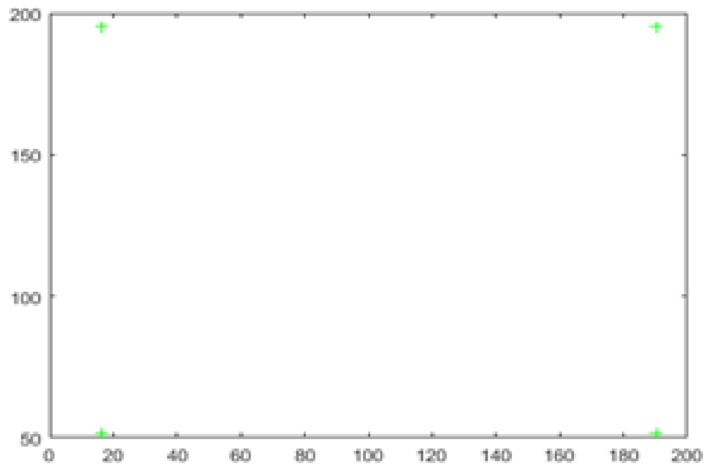


Figure 5: Four corners of the LEA found using Harris corner detection.

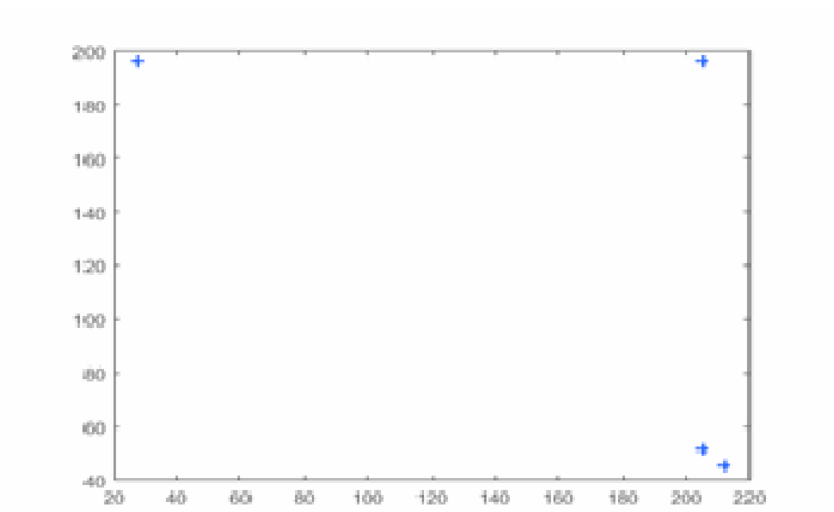


Figure 6: Harris detector finding difficult to find four corner points.

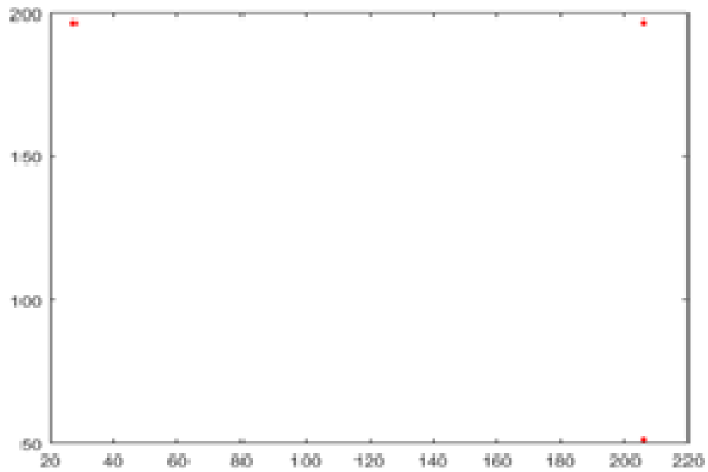


Figure 7: Combining two points which is nearer using tolerance.

arrangement was placed in the form of rectangle with black outer boundary and white inner boundary, thus improving the performance of the Harris edge-corner detection. For denoting the corner value the aggregate of the variations that was doubled is characterized as in (1).

$$E(u, v) = \sum_{(x,y)} w(x, y)[I(x + u, y + v) - I(x, y)]^2 \tag{1}$$

- E is the difference along the actual and the dispalced window.
- u is the window's displacement in the x direction
- v is the window's displacement in the y direction
- $w(x, y)$ is the window at position (x, y). This acts like a mask. Ensuring that only the desired window is used.
- I is the intensity of the image at a position (x, y)
- $I(x + u, y + v)$ is the intensity of the moved window
- $I(x, y)$ is the intensity of the original

Then, Using the Taylor series, expanding the above equation.

$$E(u, v) = B(\sum A)B' \tag{2}$$

where B is $\begin{pmatrix} u & v \end{pmatrix}$ and A is $\begin{pmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{pmatrix}$

$$M = \sum w(x, y)A \tag{3}$$

$$E(u, v) = B * M * B' \tag{4}$$

$$R = det(M) - k * (traceM)^2 \tag{5}$$

where $det(M) = \lambda_1\lambda_2$ and $traceM = \lambda_1 + \lambda_2$

If R is greater, then the obtained values are corners

MATLAB function used:

corners = detectHarrisFeatures (rgb2gray (INPUTIMAGE));

Problem Statement and the Solution provided:

After Harris corner detection algorithm, the four corners of the rectangle array have to be obtained and the mid-point of the rectangle array has to be calculated. With the help of the mid-point, the rectangle array can be tracked using the Kalman filter, which was termed as the eye of the subordinate vehicles.

The four corner points and the midpoint of the rectangle were found. Sometimes the corner detection algorithm finds it difficult to find the four corner points and where the tolerance is used to make the closest points as the whole, thus helps us to get the fourth point, and then the LEA was found using the basic rectangle geometric properties. For instance, (x1, y1) were the top left most co-ordinates of the rectangle and (x3, y3) were the bottom right most co-ordinates of the same rectangle, then the bottom left most co-ordinates will be (x1, y3) and the top right most co-ordinates will be (x3, y1).

Error in detection of the corners was a problem that arises during the corner detection phase, because the algorithm detects the corners which have the sharp change over the pixels, so in-order to solve this problem, the geometric properties were used to find the corner that was not detected. Finally through the Kalman filter, tracking results for the cases with and without the geometric properties were plotted accordingly to show the best tracking.

LEA tracking using Kalman filter:

The Kalman filter utilizes the estimation of object of the LEA in the obtained picture when the comparative assumption between the LEA and camera displaces,

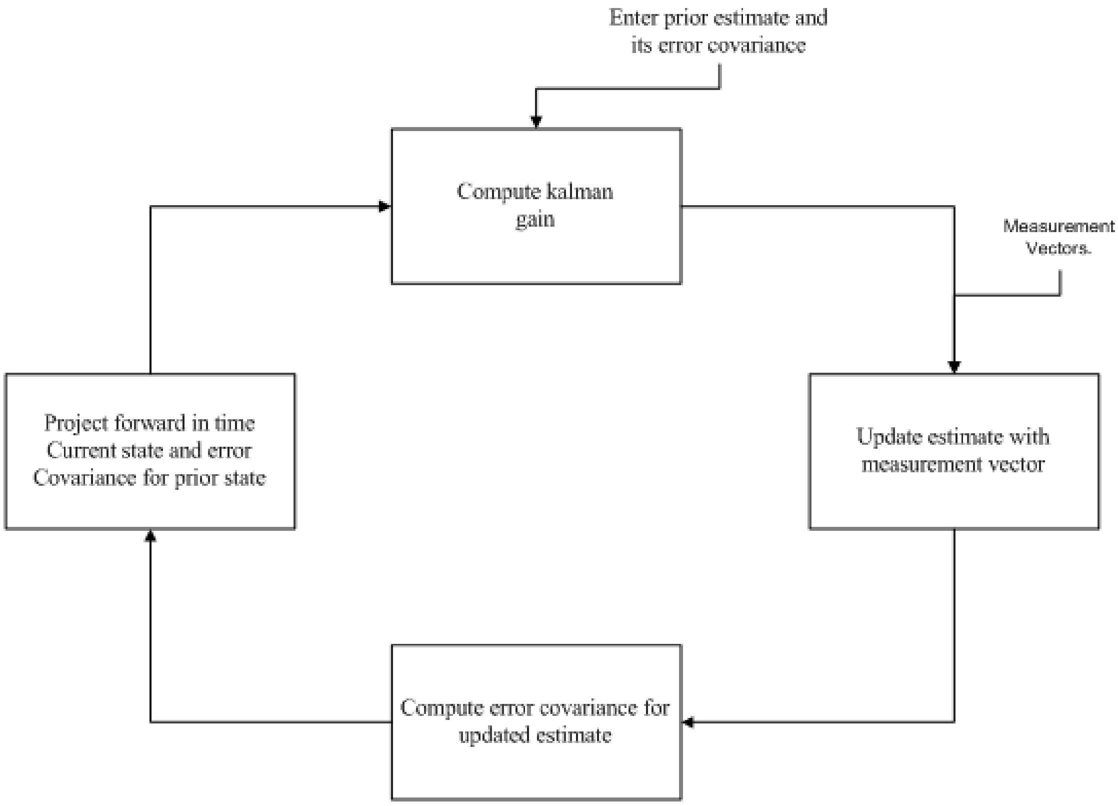


Figure 8: Kalman filter loop.

camera being the receiver. In addition to that the Kalman filter is used to track the position of the LEA in the complex background. The working of the Kalman filter is totally based on the four corner points obtained using the Harris corner detection and the midpoint of the LEA using the four corner points as per findings.

To apply Kalman filter to indicate the movement of the LEA in the obtained picture plane, we provide a midpoint of the LEA as the state vector x_k and the measure variable quantity y_k at time step k are defined as in the below Equation 6. Figure 8 shows the detailed summary of the Kalman filter operation.

$$x_k = \begin{pmatrix} x & y & v_x & v_y \end{pmatrix} y_k = \begin{pmatrix} x & y \end{pmatrix} \tag{6}$$

Tracking results of moving LEA:

The tracking results of each and every co-ordinates for each cases, for the first 20 iterations were shown below from figure 9 to figure 12.

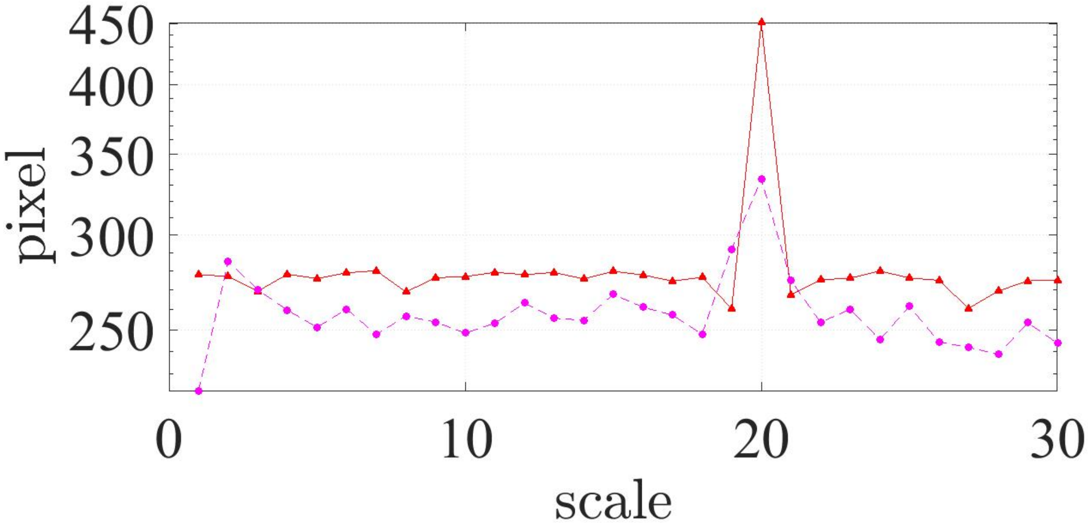


Figure 9: Center co-ordinate (x): Kalman filter tracking for case 1.

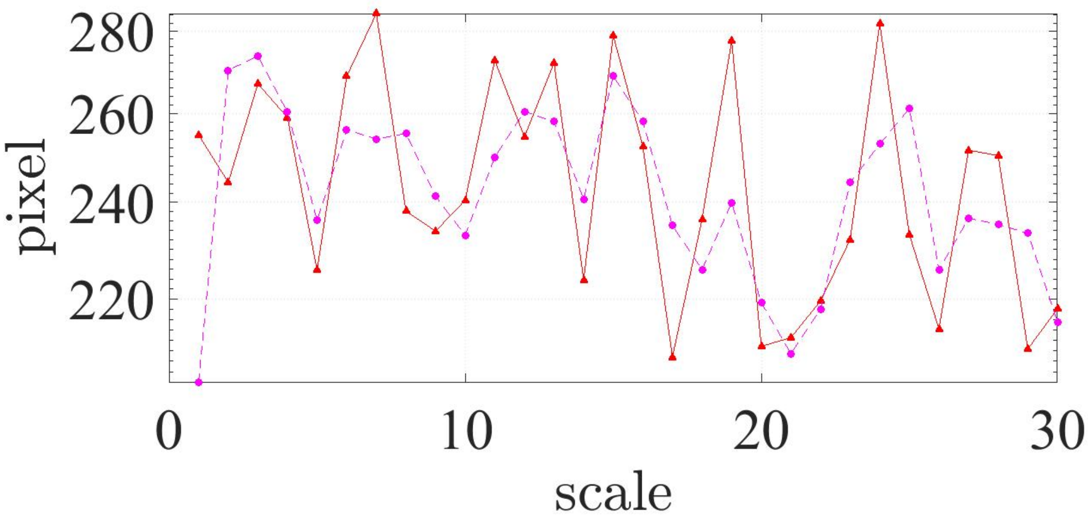


Figure 10: Center co-ordinate (y): Kalman filter tracking for case 1.

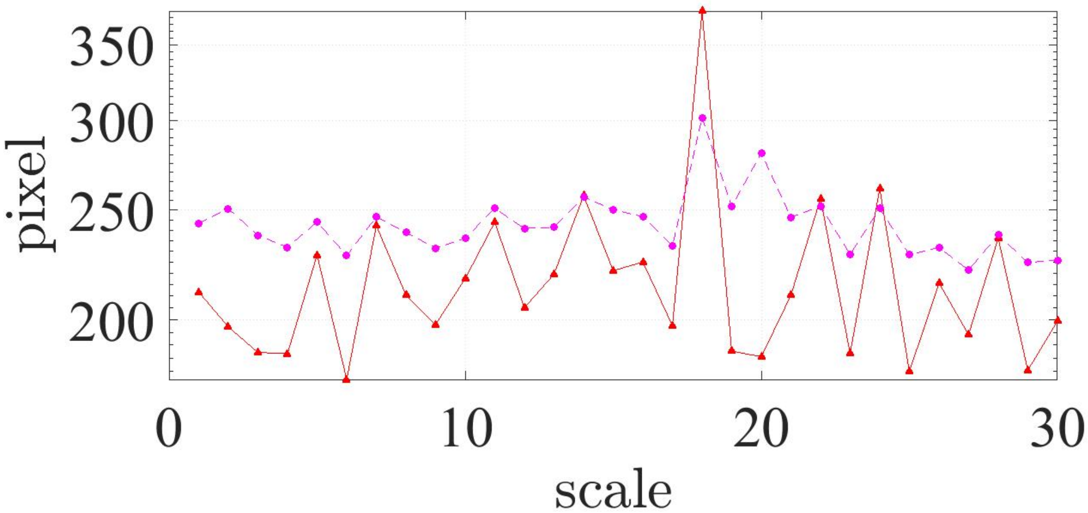


Figure 11: Center co-ordinate (x): Kalman filter tracking for case 2.

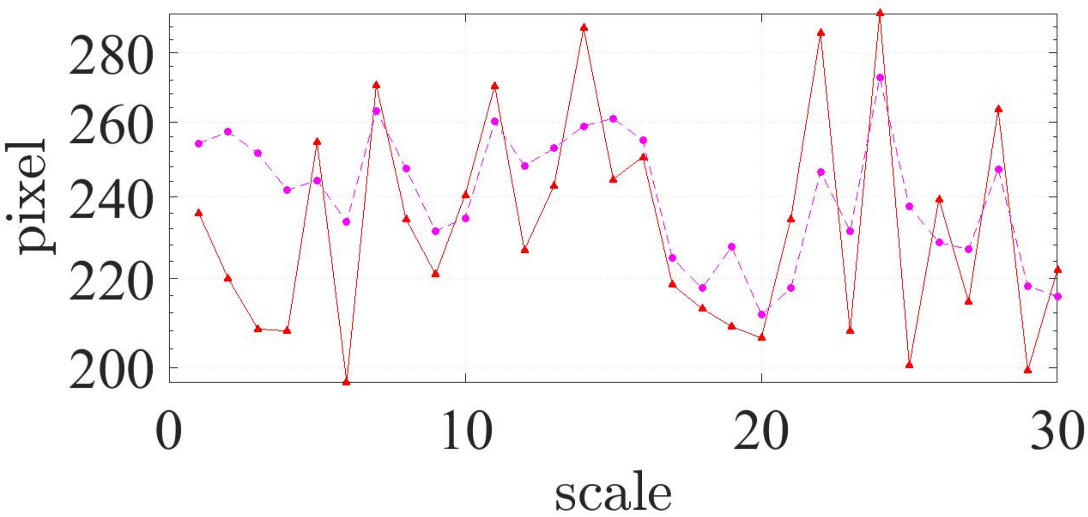


Figure 12: Center co-ordinate (y): Kalman filter tracking for case 2.

Conclusion

The project is devised using a LEA location and pursual technique for the visual-MIMO framework using image analysing system that can be utilized for different applications. For building a reliable quality of LEA location, we chose the ROI that is the LEA, utilizing Harris corner detection for locating and kalman filter for tracking, with a composite foundation and background.

This project is mainly focusing on obtaining the LEA location and tracking the same for the seamless communications. Moreover the system is using LEDs and optical communications so the cost of design was simple and effective rather compared with RF wireless communications.

MATLAB CODE

```
clc;
close all;
clear all;

%------------------------initialization-------------------------------------
No_Of_Imgs = 20;

%------------------------iterations used------------------------------------
iteration_1=1;
i_1 = 1;

for s = 1:No_Of_Imgs

    var =0.5;

%%---------------Implementation with geomentry properties------------------------------
    Im=imread('canv.jpg');
    Imbg=imread('sasas.jpg');
    Imbgsize=size(Imbg);

    %---------------scaling factor-----------------------------------------
    scaleK(s)=0.8+var*rand;
    position =[20 20];

    %---------------implementing scaling in the image----------------------
    Imfg=imresize(Im,scaleK(s));
    Imfgsize=size(Imfg);
    Imbg(position(1):position(1)+Imfgsize(1)-1,position(2):position(2)+Imfgsize(2)-1,:)=Imfg;
    figure;
    imshow(Imbg);

    %----------------------------Harris corner implementation-------------
    corners = detectHarrisFeatures(rgb2gray(Imbg));
    a = corners.selectStrongest(4);

    %--------------locating the strong points------------------------------
    b = a.Location;
    hold on;
    figure,
    plot(a);
    hold on;
    b

%----calculating the tolerence and proceeding to the geometry properties---
    tol=0.5;
    C = uniquetol(b,tol,'ByRows',true)
    figure;
    plot(C(:,1), C(:,2), '.r', 'MarkerSize', 10)
   [rows cols] = size(C);

%-----calculation to find the center of the four strong points------------
    cdata = (C(:,[1 1]));
```

```
    cdata = cdata';
     cdata
     c1 = min(min(cdata));
     c2 = c1;
     c4 = max(max(cdata));
     c3 = c4;
     ddata = (C(:,[2 2]));
     ddata = ddata';
     ddata
     d1 = min(min(ddata));
     d2 = d1;
     d4 = max(max(ddata));
     d3 = d4;

     %--------------detecting the center point-----------------------------
     xc = ((d2/2)+(c1/2));
     yc = ((d1/2) + (c4/2));
    detected_center_point = [xc yc];
    detected_center_point

%----------------kalman filter implementation------------------------------
  [xh, yh] = TrackKalman(xc, yc)
  figure,
  hold on;
  plot(xc, yc, 'y*')
  hold on;
  plot(xh, yh, 'bs')
  pause(0);

  Xcsaved(:, s) = [xc yc];
  Xhsaved(:, s) = [xh yh];
  Xcsaved(:, s) = [xc yc]';
  Xhsaved(:, s) = [xh yh]';

%%-----------------Implementation without geomentry properties-----------------------------
I=imread('canv.jpg');
Isize=size(I);
Ibg=imread('sasas.jpg');
Ibgsize=size(Ibg);

%------------------scaling factor-----------------------------------------
scaleK(s)=0.8+var*rand;
position =[20 20];

%------------------implementing scaling in the image-----------------------
Ifg=imresize(I,scaleK(s));
Ifgsize=size(Ifg);
Ibg(position(1):position(1)+Ifgsize(1)-1,position(2):position(2)+Ifgsize(2)-1,:)=Ifg;
figure;
imshow(Ibg);

%-----------------------------Harris corner implementation-----------------
corners = detectHarrisFeatures(rgb2gray(Ibg));
%[features, valid_corners] = extractFeatures(recovered, corners);

%--------------locating the strong points------------------------------
```

```
q = corners.selectStrongest(4);
%---------------locating the strong points-----------------------------
 r = q.Location;
 hold on;
 figure,
 plot(q);
 hold on;
 r

%-------calculation to find the center of the four strong points------------
     xwdata = (r(:,[1 1]));
     xwdata = xwdata';
     xwdata
     xw1 = min(min(xwdata));
     xw2 = max(max(xwdata));
     ywdata = (r(:,[2 2]));
     ywdata = ywdata';
     ywdata
     yw1 = min(min(ywdata));
     yw2 = max(max(ywdata));
     %I=imcrop(Ibg,[x1 y1 width height]);

%--------------detecting the center point----------------------------------
     xwc = ((yw2/2)+(xw1/2));
     ywc = ((yw1/2) + (xw2/2));
     detected_center_pointw = [xwc ywc];
     detected_center_pointw

%----------------kalman filter implementation------------------------------
  [xl, yl] = TrackKalman(xwc, ywc)
  figure;
   hold on;
  plot(xwc, ywc, 'y*')
  hold on;
  plot(xl, yl, 'bs')

  pause(0);

  Xpwsaved(:, s) = [xwc ywc];
  Xlsaved(:, s) = [xl yl];
  Xpwsaved(:, s) = [xwc ywc]';
  Xlsaved(:, s) = [xl yl]';


  iteration_1=iteration_1+1;
end

%-----plotting the track points of the kalman filter for both cases-------
figure,
hold on;
plot(Xcsaved(1,:), Xcsaved(2,:), '*');
hold on;
plot(Xhsaved(1,:), Xhsaved(2,:), 's');
hold on;
figure,
hold on;
```

```
plot(Xpwsaved(1,:), Xpwsaved(2,:), '*');
hold on;
plot(Xlsaved(1,:), Xlsaved(2,:), 's');
hold on;


%close all;

%-----------comparing the tracking plots for both cases--------------------
scale = 1:1:No_Of_Imgs;
figure;
posi1_Plot=semilogy(scale,Xcsaved(1,:),'marker','^','markersize',6,'markerfacecolor','r','lin
estyle','-','color','r');
hold on;

esti1_Plot=semilogy(scale,Xhsaved(1,:),'marker','o','markersize',6,'markerfacecolor','m','lin
estyle','--','color','m');
hold on;

grid on
xlabel('$$\mathrm{scale}$$','FontName','Times New
Roman','FontSize',50,'Interpreter','latex');
ylabel('$$\mathrm{pixel}$$','FontName','Times New
Roman','FontSize',50,'Interpreter','latex');
%legend([posi1_Plot esti1_Plot],'actual position','estimated position',+2);
set(gca,'FontSize',50,'XGrid','on','YGrid','on','GridLineStyle',':','MinorGridLineStyle','non
e','FontName','Times New Roman');

scale = 1:1:No_Of_Imgs;
figure;
posi2_Plot=semilogy(scale,Xcsaved(2,:),'marker','^','markersize',6,'markerfacecolor','r','lin
estyle','-','color','r');
hold on;

esti2_Plot=semilogy(scale,Xhsaved(2,:),'marker','o','markersize',6,'markerfacecolor','m','lin
estyle','--','color','m');
hold on;

grid on
xlabel('$$\mathrm{scale}$$','FontName','Times New
Roman','FontSize',50,'Interpreter','latex');
ylabel('$$\mathrm{pixel}$$','FontName','Times New
Roman','FontSize',50,'Interpreter','latex');
%legend([posi2_Plot esti1_Plot],'actual position','estimated position',+2);
set(gca,'FontSize',50,'XGrid','on','YGrid','on','GridLineStyle',':','MinorGridLineStyle','non
e','FontName','Times New Roman');


scale = 1:1:No_Of_Imgs;
figure;
posi3_Plot=semilogy(scale,Xpwsaved(1,:),'marker','^','markersize',6,'markerfacecolor','r','li
nestyle','-','color','r');
hold on;

esti3_Plot=semilogy(scale,Xlsaved(1,:),'marker','o','markersize',6,'markerfacecolor','m','lin
estyle','--','color','m');
```

```
hold on;

grid on
xlabel('$$\mathrm{scale}$$','FontName','Times New
Roman','FontSize',50,'Interpreter','latex');
ylabel('$$\mathrm{pixel}$$','FontName','Times New
Roman','FontSize',50,'Interpreter','latex');
%legend([posi1_Plot esti1_Plot],'actual position','estimated position',+2);
set(gca,'FontSize',50,'XGrid','on','YGrid','on','GridLineStyle',':','MinorGridLineStyle','non
e','FontName','Times New Roman');

scale = 1:1:No_Of_Imgs;
figure;
posi4_Plot=semilogy(scale,Xpwsaved(2,:),'marker','^','markersize',6,'markerfacecolor','r','li
nestyle','-','color','r');
hold on;

esti4_Plot=semilogy(scale,Xlsaved(2,:),'marker','o','markersize',6,'markerfacecolor','m','lin
estyle','--','color','m');
hold on;

grid on
xlabel('$$\mathrm{scale}$$','FontName','Times New
Roman','FontSize',50,'Interpreter','latex');
ylabel('$$\mathrm{pixel}$$','FontName','Times New
Roman','FontSize',50,'Interpreter','latex');
%legend([posi2_Plot esti1_Plot],'actual position','estimated position',+2);
set(gca,'FontSize',50,'XGrid','on','YGrid','on','GridLineStyle',':','MinorGridLineStyle','non
e','FontName','Times New Roman');
```

```
Warning: Image is too big to fit on screen;
displaying at 67%

b =

   36.4001   37.4323
  447.7591   37.5271
   36.5300  402.0515
  468.4582   20.5533



C =

   36.4001   37.4323
   36.5300  402.0515
  447.7591   37.5271



cdata =

   36.4001   36.5300  447.7591
   36.4001   36.5300  447.7591



ddata =
```

```
   37.4323  402.0515   37.5271
   37.4323  402.0515   37.5271


detected_center_point =

  276.9162  242.5957


xh =

  221.7536


yh =

  194.2699

Warning: Image is too big to fit on screen;
displaying at 67%

r =

   38.4546   39.5095
  497.8367   39.5334
   38.4410  446.6477
  521.4609   20.5807


xwdata =

   38.4546  497.8367   38.4410  521.4609
   38.4546  497.8367   38.4410  521.4609


ywdata =

   39.5095   39.5334  446.6477   20.5807
   39.5095   39.5334  446.6477   20.5807


detected_center_pointw =

  242.5443  271.0208


xl =

  265.8558


yl =

  276.2022

Warning: Image is too big to fit on screen;
displaying at 67%
```

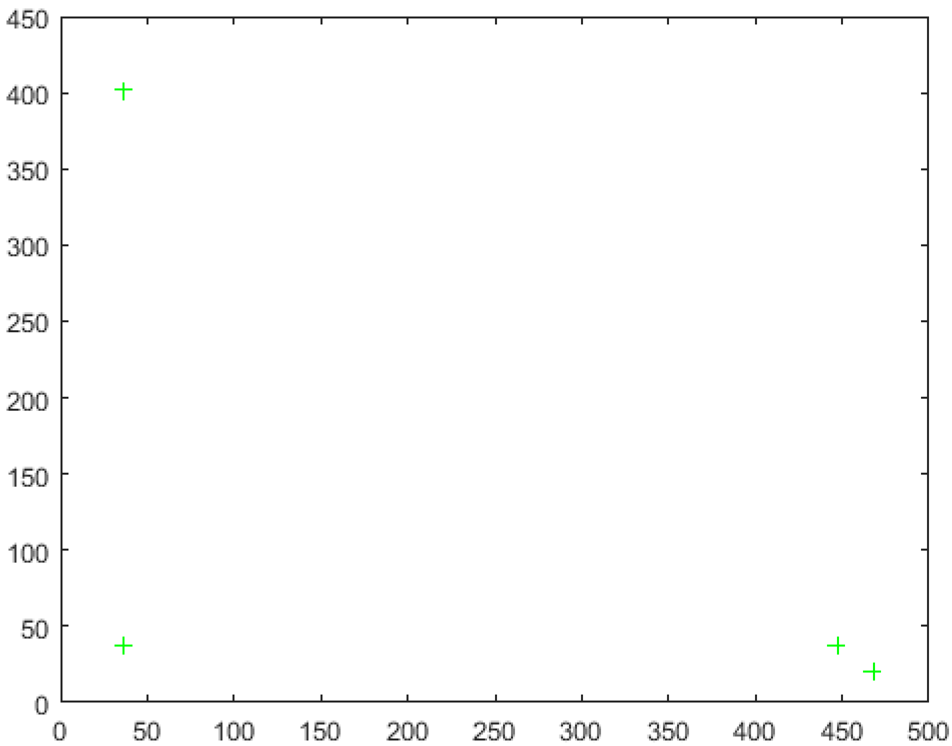
```
b =

   39.5033  474.5107
   39.5417   40.7330
  529.0895  474.5603
  553.6090   20.4496


C =

   39.5033  474.5107
   39.5417   40.7330
  529.0895  474.5603
  553.6090   20.4496


cdata =

   39.5033   39.5417  529.0895  553.6090
   39.5033   39.5417  529.0895  553.6090


ddata =

  474.5107   40.7330  474.5603   20.4496
  474.5107   40.7330  474.5603   20.4496


detected_center_point =

  269.9764  287.0293


xh =

  291.6052


yh =

  311.5100

Warning: Image is too big to fit on screen;
displaying at 67%

r =

   33.5500   34.4568
   33.5169  329.7871
  384.4556   20.5040
  367.0783   34.4527


xwdata =

   33.5500   33.5169  384.4556  367.0783
```

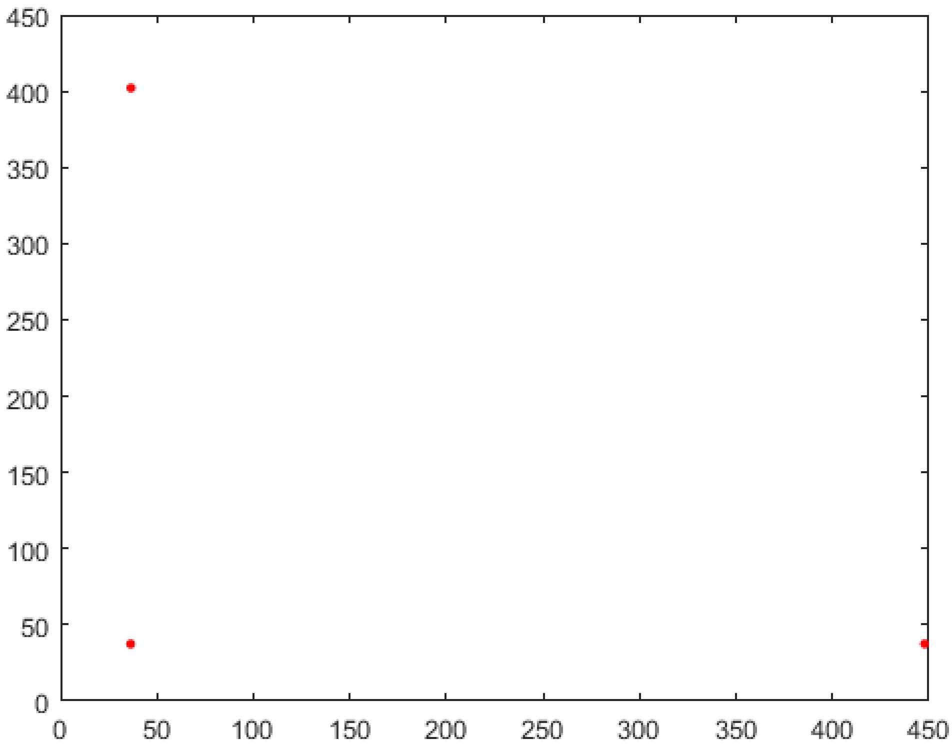
```
   33.5500   33.5169  384.4556  367.0783


ywdata =

   34.4568  329.7871   20.5040   34.4527
   34.4568  329.7871   20.5040   34.4527


detected_center_pointw =

  181.6520  202.4798


xl =

  243.7142


yl =

  269.1625

Warning: Image is too big to fit on screen;
displaying at 67%

b =

  455.4810  408.7960
  455.4872   37.7879
  476.4535   20.5546
   20.5564  425.4581


C =

   20.5564  425.4581
  455.4810  408.7960
  455.4872   37.7879


cdata =

   20.5564  455.4810  455.4872
   20.5564  455.4810  455.4872


ddata =

  425.4581  408.7960   37.7879
  425.4581  408.7960   37.7879


detected_center_point =

  269.1721  246.6376
```

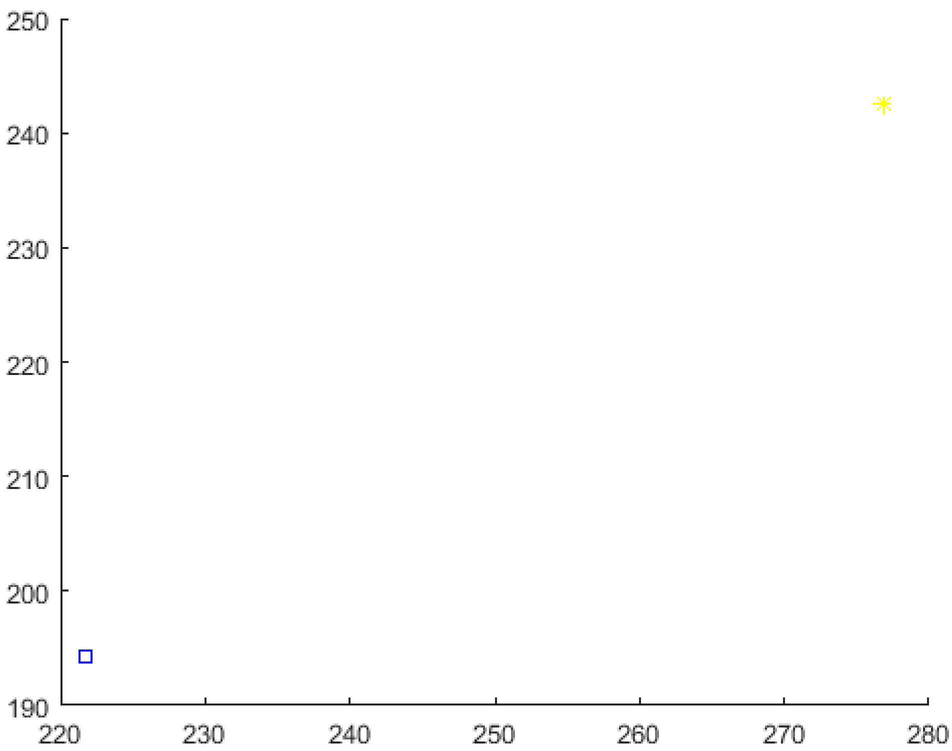
```
xh =

  264.5354


yh =

  267.8958

Warning: Image is too big to fit on screen;
displaying at 67%

r =

  435.4496  390.9664
  435.4542   37.1106
  455.4540   20.5472
   20.5454  407.4589


xwdata =

  435.4496  435.4542  455.4540   20.5454
  435.4496  435.4542  455.4540   20.5454


ywdata =

  390.9664   37.1106   20.5472  407.4589
  390.9664   37.1106   20.5472  407.4589


detected_center_pointw =

  214.0022  238.0006


xl =

  247.0860


yl =

  261.0396
```

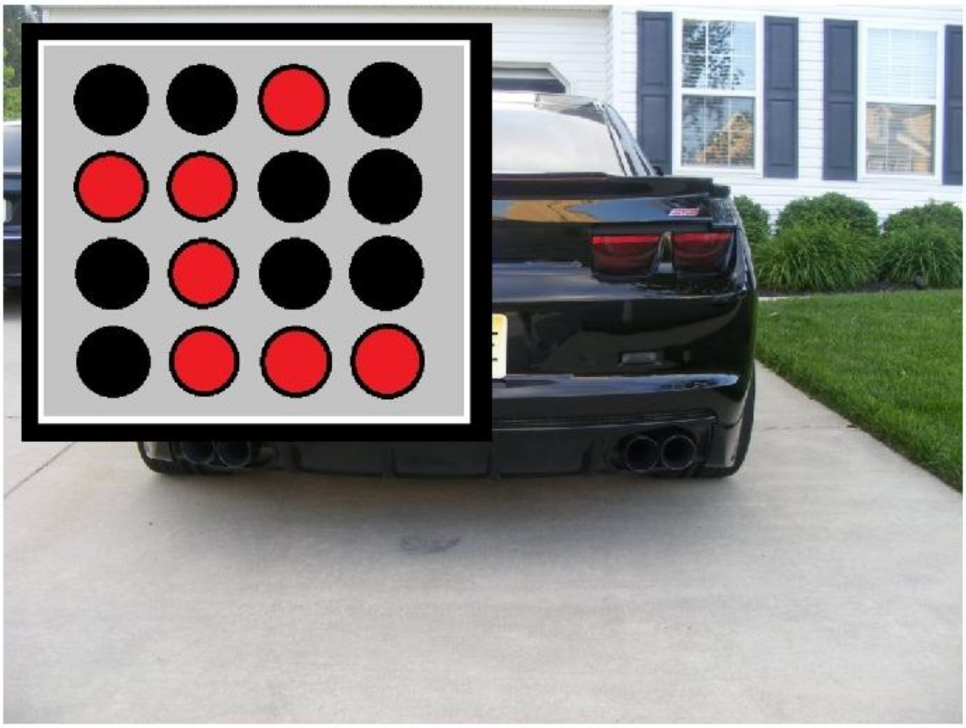
The results for the first three iterations were shown below:

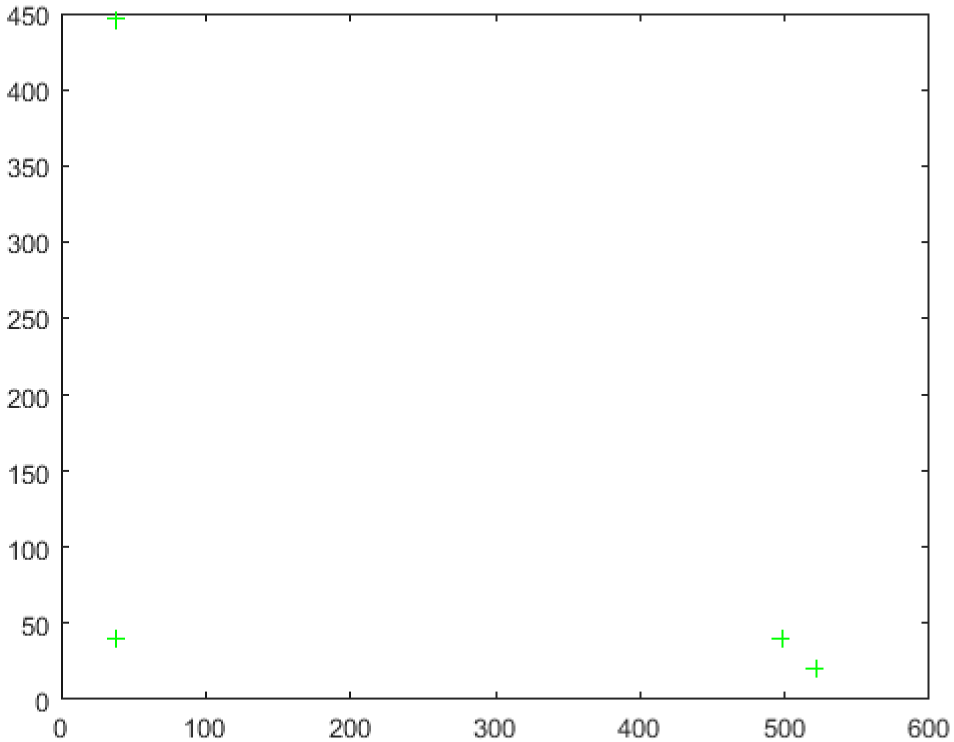

TEE


450
400
350
300
250
200
150
100
50
0
0 50 100 150 200 250 300 350 400 450 500

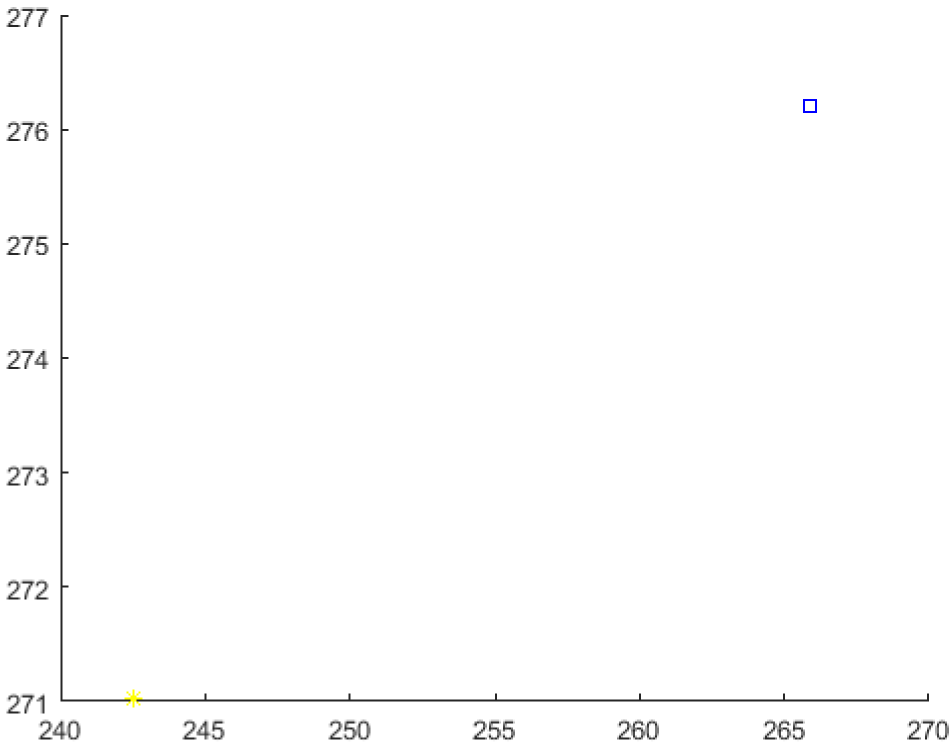

450
400
350
300
250
200
150
100
50
0
0
50
100
150
200
250
300
350
400
450


250
240
230
220
210
200
190
220
230
240
250
260
270
280

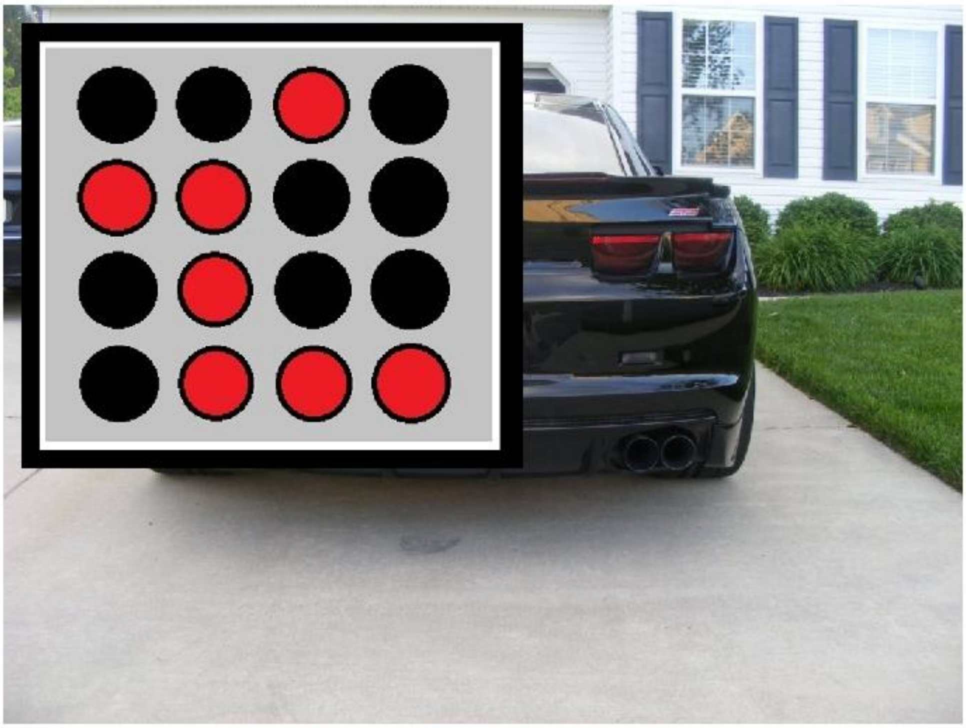


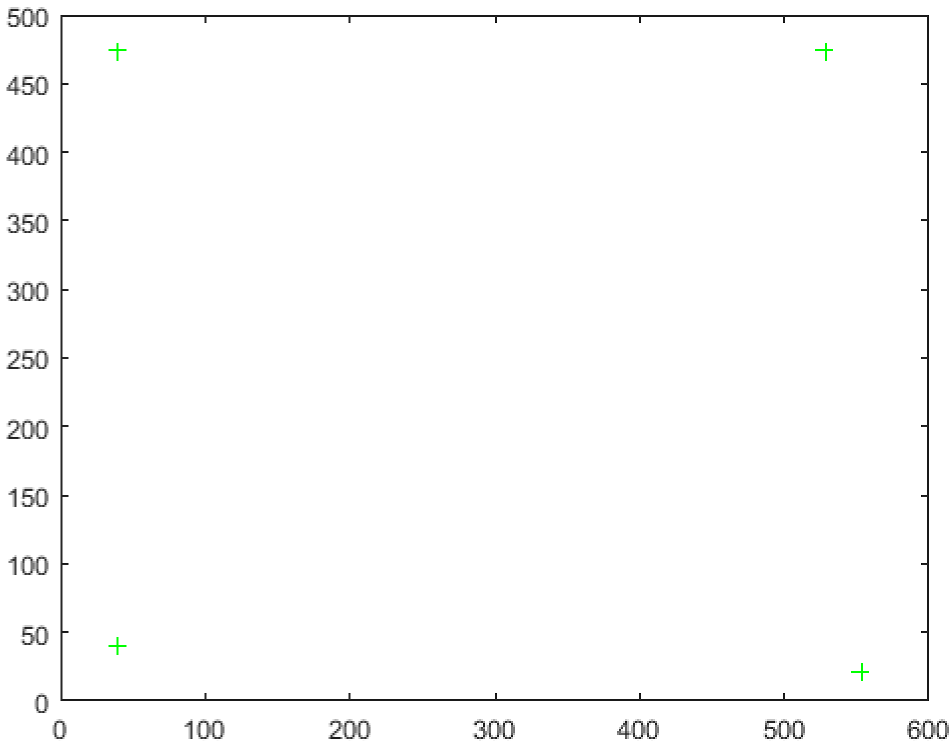


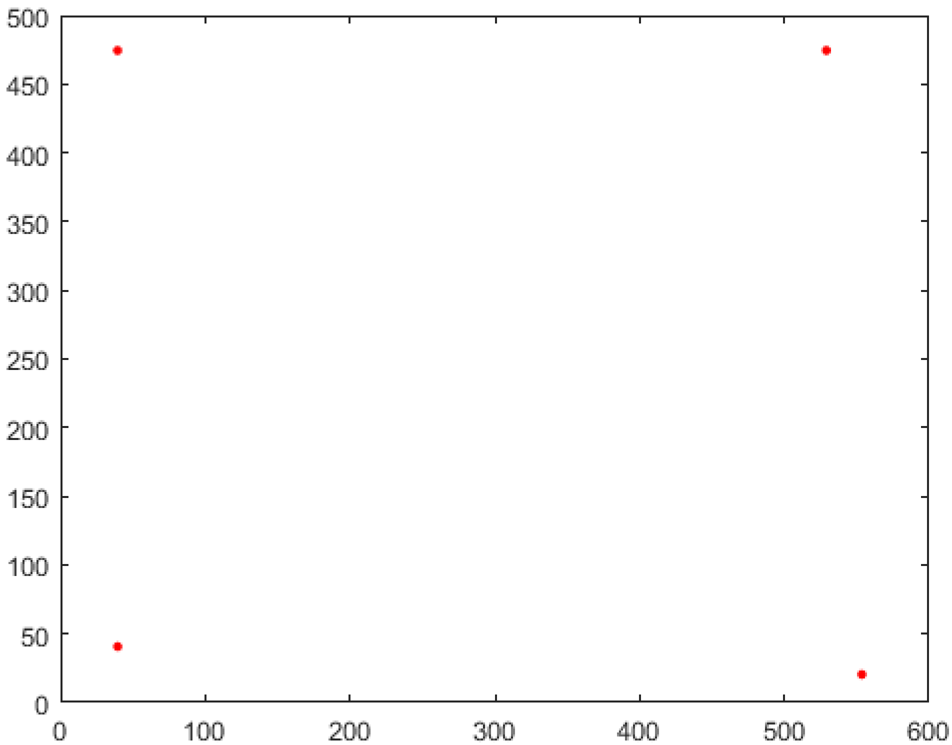
450
400
350
300
250
200
150
100
50
0
0
100
200
300
400
500
600

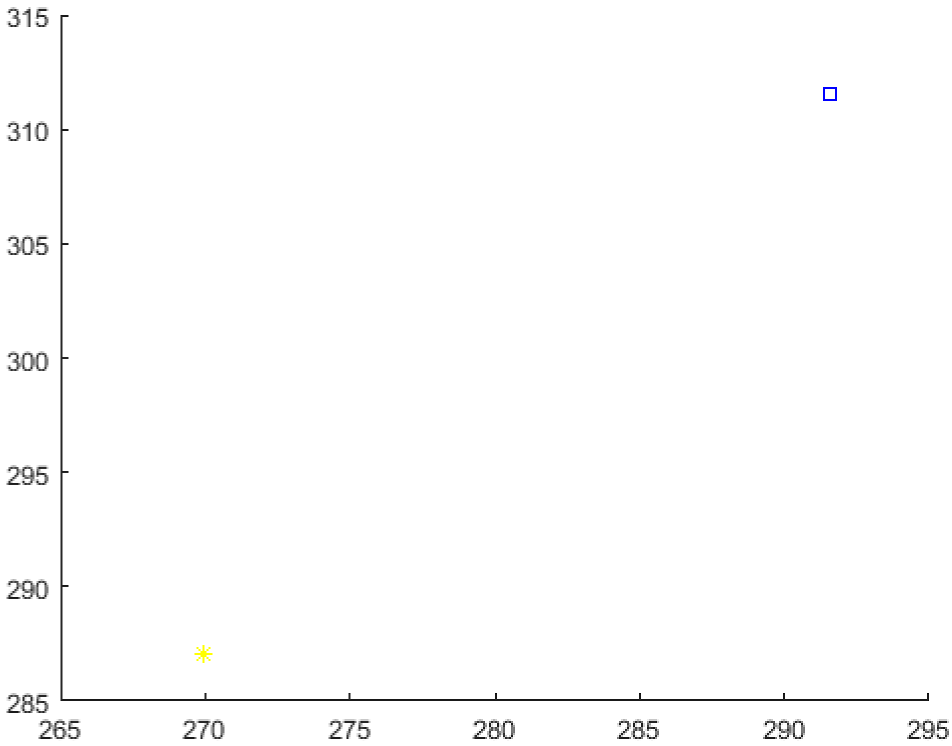


277
276
275
274
273
272
271
240
245
250
255
260
265
270

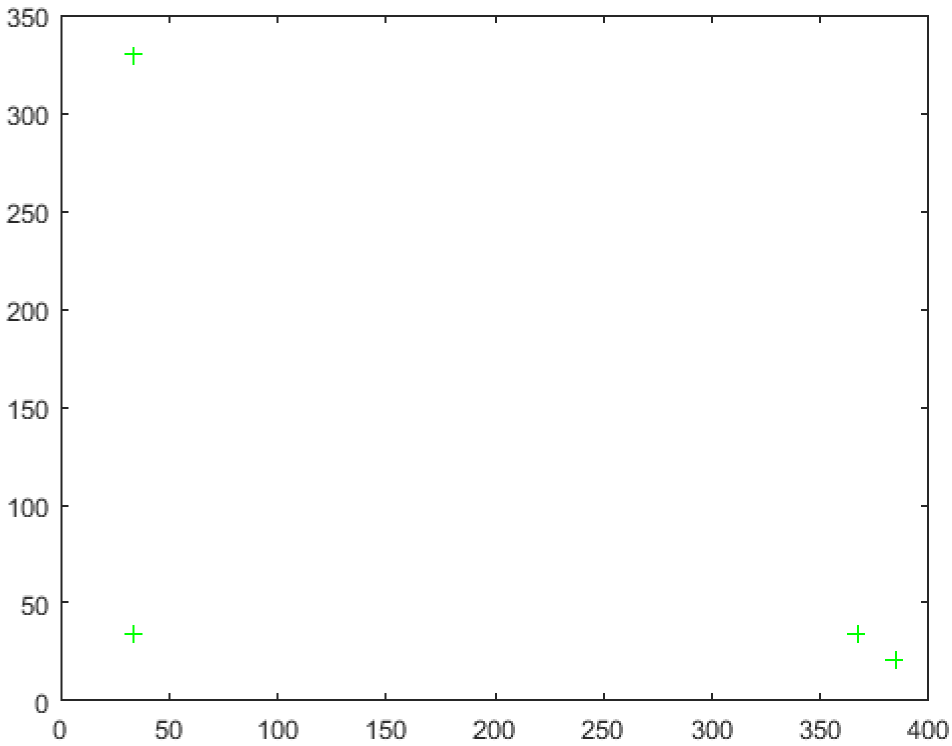


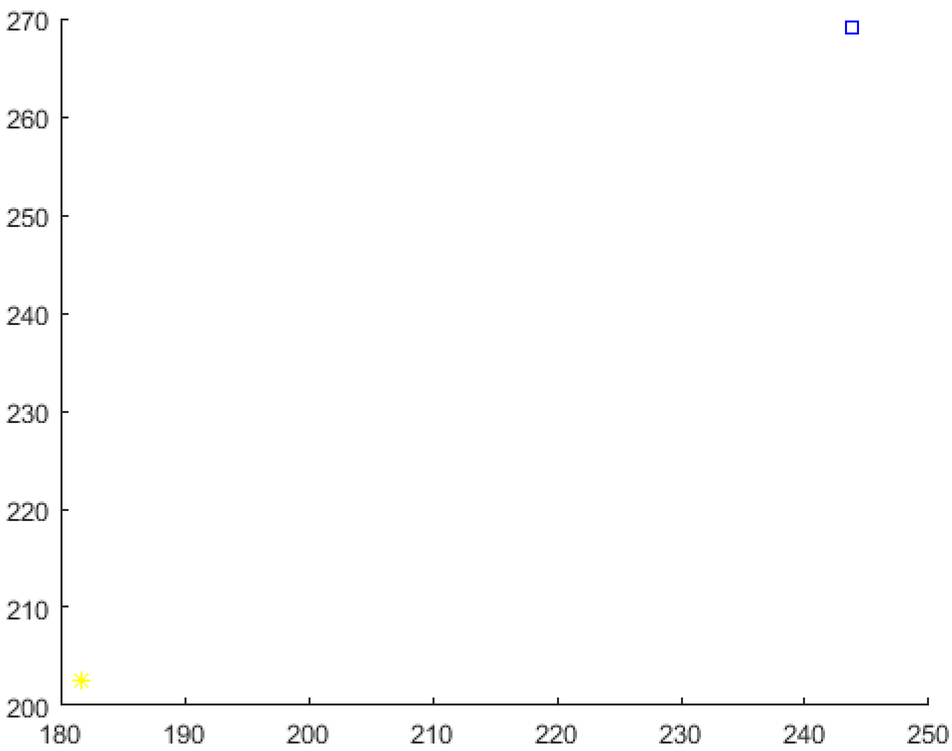

500
450
400
350
300
250
200
150
100
50
0
0
100
200
300
400
500
600

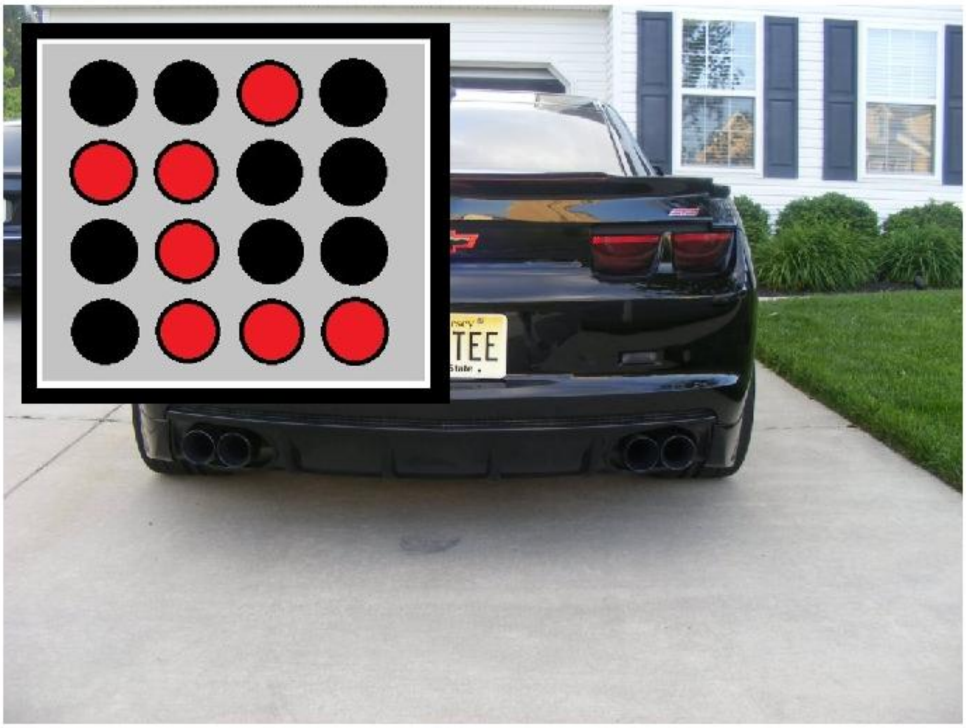

500
450
400
350
300
250
200
150
100
50
0
0
100
200
300
400
500
600

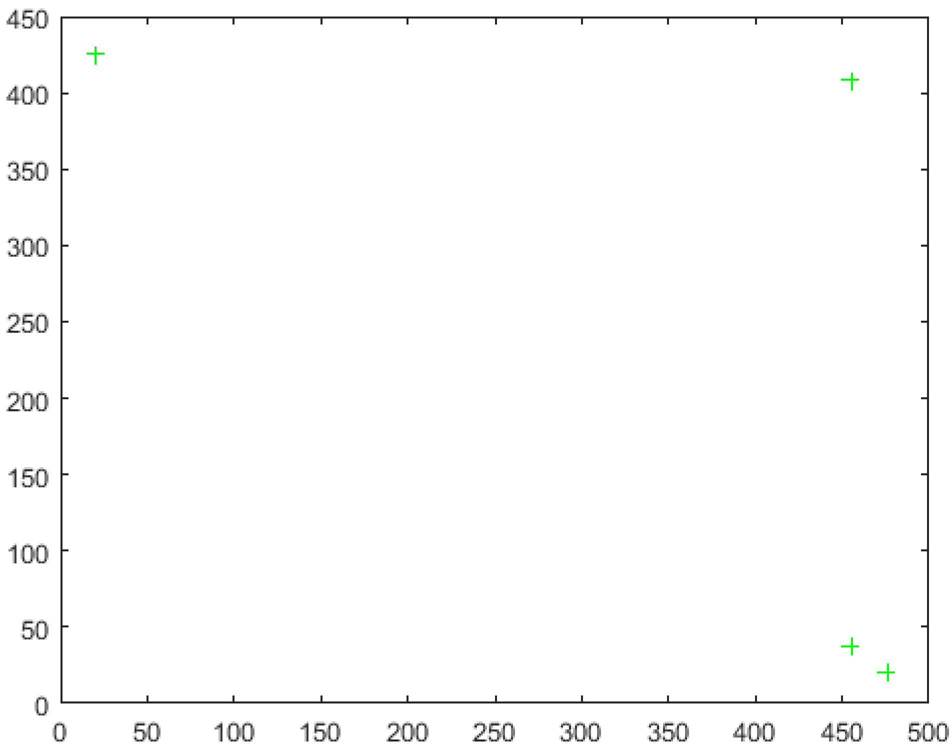

315
310
305
300
295
290
285
265
270
275
280
285
290
295


New Jersey
LADITEE
Garden State

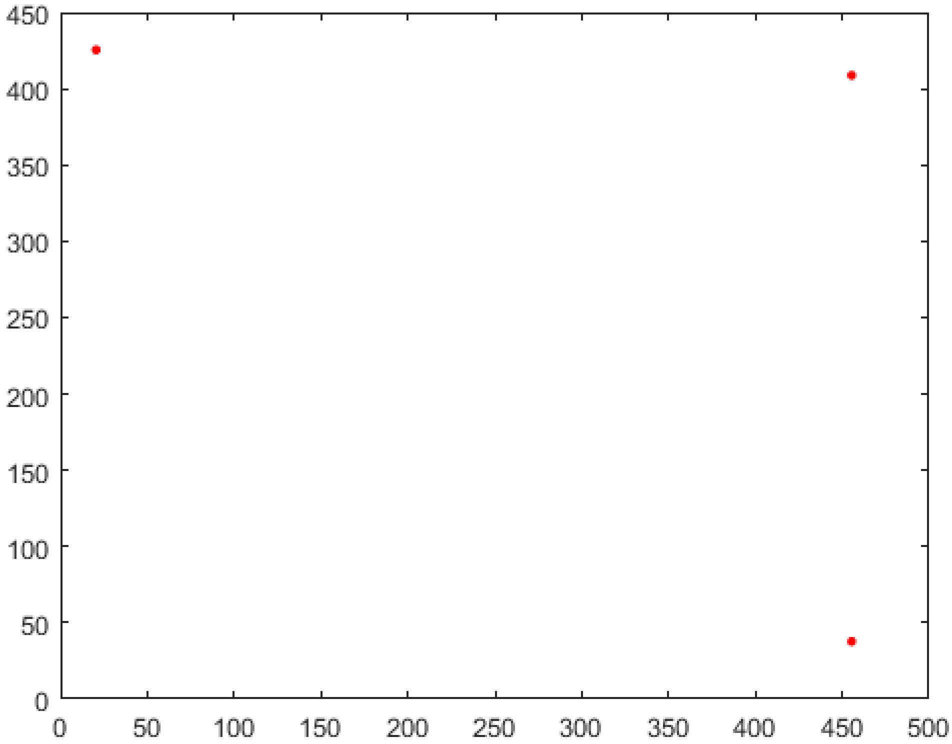

350
300
250
200
150
100
50
0
0
50
100
150
200
250
300
350
400


270
260
250
240
230
220
210
200
180
190
200
210
220
230
240
250

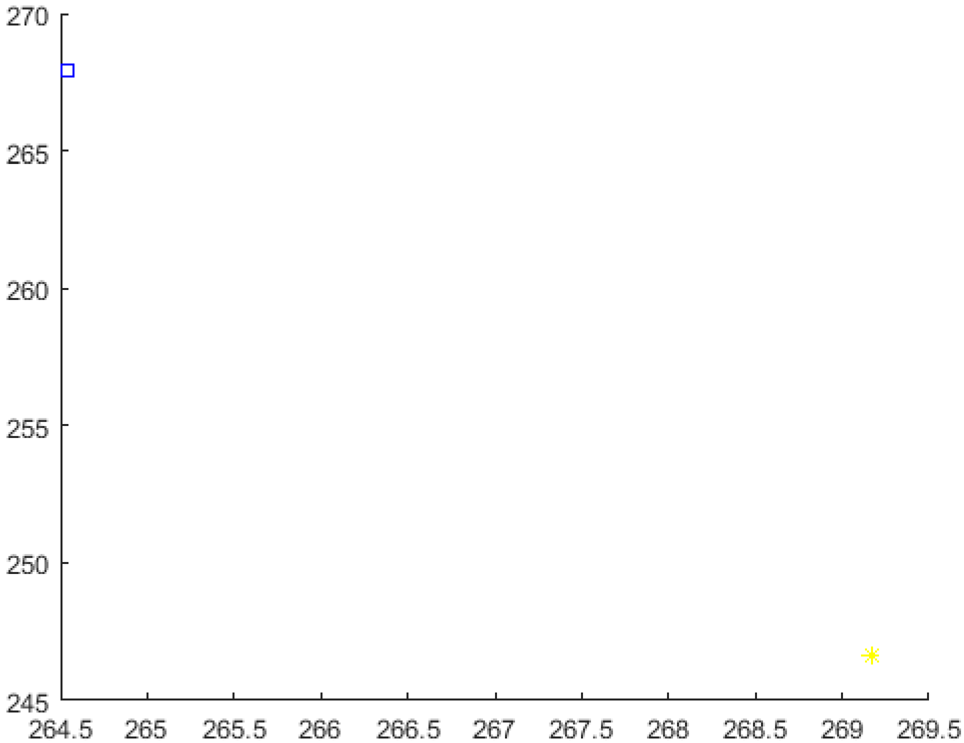




450
400
350
300
250
200
150
100
50
0
0 50 100 150 200 250 300 350 400 450 500

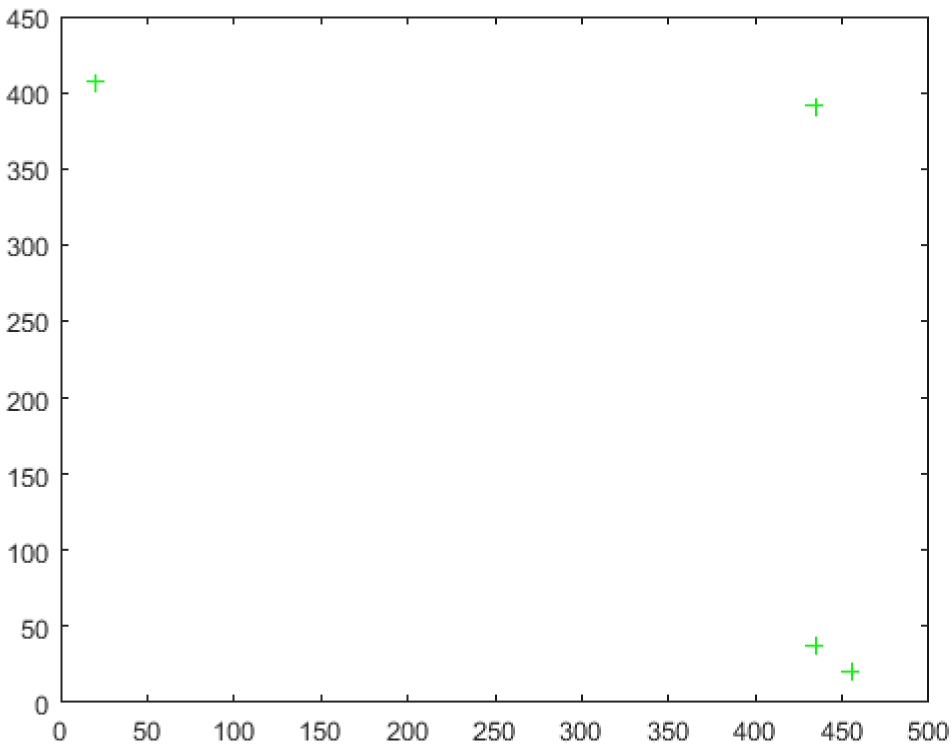


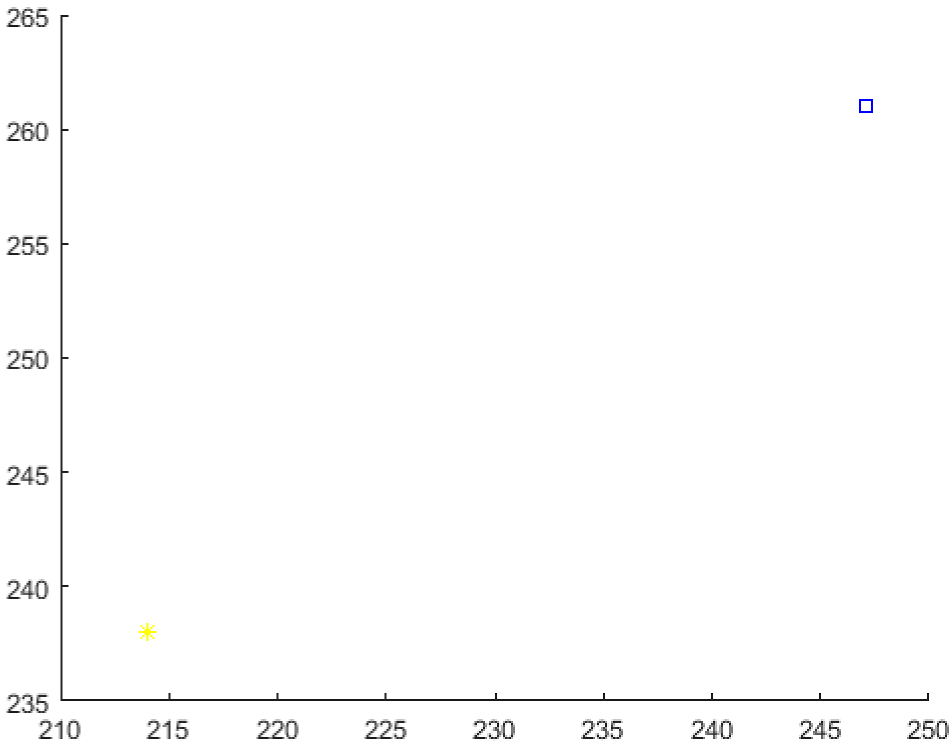
450
400
350
300
250
200
150
100
50
0
0 50 100 150 200 250 300 350 400 450 500



270
265
260
255
250
245
264.5 265 265.5 266 266.5 267 267.5 268 268.5 269 269.5


ITEE


450
400
350
300
250
200
150
100
50
0
0
50
100
150
200
250
300
350
400
450
500


265
260
255
250
245
240
235
210
215
220
225
230
235
240
245
250