APPENDIX A

SIMULATION SCENARIOS DISCRIPTION

Throughout this work, we obtained the attributes mentioned in the previous sections and we proved their utilities in the field of video surveillance based on three case studies. Non human case study, single human case study and two human case study. The coming section explains these case studies and the various scenarios in each one.

1. Single non human object

This case study consists of one scenario; a ball hanged by string and moved freely in the space [Altahir A. Altahir et al, 2007].

2. Single human

This case study consists of three scenarios; the first scenario is a single agent walking, the second scenario is a single agent walking and then he stopped and after that he started to walk again the third scenario is a single agent running [Altahir A. Altahir et al, 2008a].

3. Two human

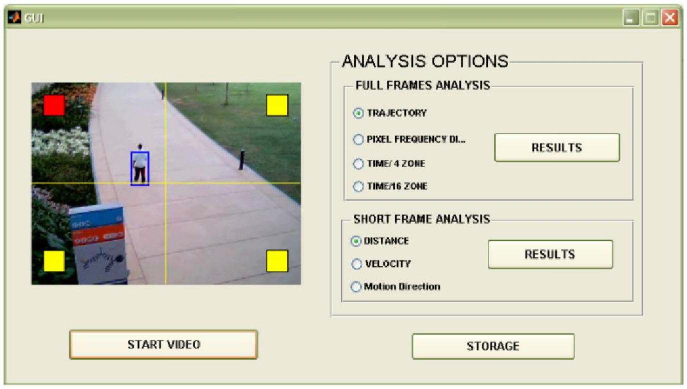
This case study consists of three scenarios; the first scenario is two agents walking slowly in the same direction. The second scenario is two agents walking faster (compared to the first scenario) in the same direction and finally the third scenario is consider two agents walking toward each other [Altahir A. Altahir et al, 2008d].

APPENDIX B

THE GRAPHIC USER INTERFACE

1. The Main Window

The main window shown below illustrates the graphic user interface for observing human activities from video streams. The main window consists of monitor dedicated for displaying the current activities and control panel dedicated for the analysis options.



The Main Graphic User interface window

2. The Monitor

The core element in the graphic user interface is the monitor and the purpose of the monitor is to display the current human activities and to assists the human operators caching abnormal events by highlighting the active section in the image plane.

3. The Analysis Options

The analysis options provides a detailed description for the current motion aspect based on full frame, inter frame and region based analysis. It also provides the expected storage rate after the reduction.

APPENDIX C

MATLAB CODES

1. Background Subtraction Code

```
function [STATS,M,BG, zT] = BwIm(CF,BG,alpha,zT)
CFgray = imabsdiff(CF,BG);
for i = 1:240
 for j = 1:320
    if CFgray(i,j)>= 20
      BG(i,j) = alpha*CFgray(i,j) + (1-alpha)*BG(i,j);
      CFgray(i,j) = 255;
    else
      CFgray(i,j) = 0;
    end
 end
end
BW = im2bw(CFgray,0.5);
se = strel('disk',3);
L = bwlabel(BW);
STATS = regionprops(L,'Area');
idx = find([STATS.Area] > 50);
BW = ismember(L,idx);
cF = imclose( BW,se);
BW = imfill(cF,'holes');
L = bwlabel(BW);
L1(1:240,:) = L(240:-1:1,:);
zT = zT + L1;
STATS = regionprops(L,'basic');
M = length(STATS);
```

2. Tracking Code

```
function BBox(CF,x,y,w,h,x1,y1,i,handles)
t(:,1) = 160.*(ones(240,1));  t(:,2) = (1:240);
u(:,1) = (1:320);  u(:,2) = 120.*(ones(320,1));
switch handles.l(i)
   case 1
   imshow(CF)
   hold on
   rectangle('Position',[x y w h],'LineWidth',2,'EdgeColor','b');
   rectangle('Position',[15 15 25 25],'FaceColor','r');
   rectangle('Position',[15 200 25 25],'FaceColor','y');
   rectangle('Position',[280 15 25 25],'FaceColor','y');
   rectangle('Position',[280 200 25 25],'FaceColor','y');
   plot(t(:,1),t(:,2), 'y-');  plot(u(:,1),u(:,2), 'y-');
   hold off
   getframe;
   case 3
   imshow(CF)
   hold on
   rectangle('Position',[x y w h],'LineWidth',2,'EdgeColor','b');
   rectangle('Position',[15 15 25 25],'FaceColor','y');
   rectangle('Position',[15 200 25 25],'FaceColor','r');
   rectangle('Position',[280 15 25 25],'FaceColor','y');
   rectangle('Position',[280 200 25 25],'FaceColor','y');
   plot(t(:,1),t(:,2), 'y-');  plot(u(:,1),u(:,2), 'y-');
   hold off
   getframe;
   case 2
   imshow(CF)
   hold on
   rectangle('Position',[15 15 25 25],'FaceColor','y');
   rectangle('Position',[15 200 25 25],'FaceColor','y');
   rectangle('Position',[280 15 25 25],'FaceColor','r');
   rectangle('Position',[280 200 25 25],'FaceColor','y');
   rectangle('Position',[x y w h],'LineWidth',2,'EdgeColor','b');
   plot(t(:,1),t(:,2), 'y-');   plot(u(:,1),u(:,2), 'y-');
   hold off
   getframe;
    case 4
   imshow(CF)
   hold on
   rectangle('Position',[15 15 25 25],'FaceColor','y');
   rectangle('Position',[15 200 25 25],'FaceColor','y');
   rectangle('Position',[280 15 25 25],'FaceColor','y');
   rectangle('Position',[280 200 25 25],'FaceColor','r');
   rectangle('Position',[x y w h],'LineWidth',2,'EdgeColor','b');
   plot(t(:,1),t(:,2), 'y-');   plot(u(:,1),u(:,2), 'y-');
   hold off
   getframe;
   otherwise
   imshow(CF)
   hold on
   rectangle('Position',[15 15 25 25],'FaceColor','y');
   rectangle('Position',[15 200 25 25],'FaceColor','y');
   rectangle('Position',[280 15 25 25],'FaceColor','y');
   rectangle('Position',[280 200 25 25],'FaceColor','y');
   plot(t(:,1),t(:,2), 'y-');  plot(u(:,1),u(:,2), 'y-');
   hold off
   getframe;
end
```

3. The Timing Code

```
function handles = getT(handles)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x1 = ceil(nonzeros(handles.Cen(:,1)));
y1 = ceil(nonzeros(handles.Cen(:,2)));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tR1 = ismember(x1,1:80);
tR2 = ismember(x1,81:160);
tR3 = ismember(x1,161:240);
tR4 = ismember(x1,241:320);
tC1 = ismember(y1,1:60);
tC2 = ismember(y1,61:120);
tC3 = ismember(y1,121:180);
tC4 = ismember(y1,181:240);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T11 = sum(tR1.*tC1)/30;
T12 = sum(tR2.*tC1)/30;
T13 = sum(tR1.*tC2)/30;
T14 = sum(tR2.*tC2)/30;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T21 = sum(tR3.*tC1)/30;
T22 = sum(tR4.*tC1)/30;
T23 = sum(tR3.*tC2)/30;
T24 = sum(tR4.*tC2)/30;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T31 = sum(tR1.*tC3)/30;
T32 = sum(tR2.*tC3)/30;
T33 = sum(tR1.*tC4)/30;
T34 = sum(tR2.*tC4)/30;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T41 = sum(tR3.*tC3)/30;
T42 = sum(tR4.*tC3)/30;
T43 = sum(tR3.*tC4)/30;
T44 = sum(tR4.*tC4)/30;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
handles.TA = [T13,T14,T11,T12];
handles.TB = [T23,T24,T21,T22];
handles.TC = [T31,T32,T33,T34];
handles.TD = [T43,T44,T41,T42];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
handles.T = [sum(handles.TA),sum(handles.TB),sum(handles.TC),sum(handles.TD)];
```

4. Reducing the Storage capacity

```
function s = stg(handles)
v = handles.vel;
p =  handles.ps;
for i=1: length(v)
    if v(i)<= 20 && p(i) == 0
        s(i) = 10;
    elseif v(i)<= 20 && p(i)<= 75
        s(i) = 15;
    elseif v(i)<= 20 && p(i)> 75
        s(i) = 20;
    elseif v(i)>= 20 && p(i)> 75
         s(i) = 20;
    else
        s(i) = 25;
    end
end
figure,stem(1:length(s),s,'fill','b-o')
axis([0 22 0 27 ]),grid on
hold on
plot(1:length(s),s,'b-o','LineWidth',3)
hold off

figure,plot(1:length(p),p,'-bo',...
                'LineWidth',2,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor',[0.01 0.01 1],...
                'MarkerSize',8)
grid on

figure,plot(1:21,handles.vel,'-ro',...
                'LineWidth',2,...
                'MarkerEdgeColor','r',...
                'MarkerFaceColor',[1 0.01 0.01],...
                'MarkerSize',8)
grid on
```

5. The Main Code

```
function varargout = GUI(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',mfilename, ...
   'gui_Singleton',  gui_Singleton,'gui_OpeningFcn', @GUI_OpeningFcn, ...
   'gui_OutputFcn',  @GUI_OutputFcn,'gui_LayoutFcn',  [] , ...
   'gui_Callback',   []); if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1}); end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function GUI_OpeningFcn(hObject,eventdata, handles, varargin)
Frames = 529;                     mov = aviread('H1-1.avi');
alpha = 0.001;                    BG = rgb2gray(mov(1).cdata);
global zT;                        zT = zeros(240,320);
handles.A = 0;                    handles.U = 0;
tem = 0;                          handles.Cen = zeros(Frames,2);
handles.Bbox = zeros(Frames,4);       handles.d = zeros(Frames,1);
handles.l = zeros(Frames,1);          handles.pm = zeros(Frames,1);
handles.ps =zeros(fix(Frames/25),1);  handles.P = zeros(240,320);
handles.ps=zeros(fix(Frames/25),1);   handles.dist=zeros(fix(Frames/25),1);
handles.Adist=zeros(fix(Frames/25),1);handles.vel= zeros(fix(Frames/25),1);
handles.TA = zeros(1,4);              handles.TB = zeros(1,4);
handles.TC = zeros(1,4);              handles.TD = zeros(1,4);
handles.T =  zeros(1,4);              CF = zeros(240,320);

for cF = 1:1:Frames
    CF = mov(cF).cdata;
    CF = rgb2gray(CF);
    [STATS,M,BG,zT] = BwIm(CF,BG,alpha,zT);
    if M ~= 0
      for i = 1:M
```

```
          handles.Cen(cF,:)  = STATS(i).Centroid;
          handles.Bbox(cF,:) = STATS(i).BoundingBox;
          handles.d(cF) = distance('gc',[handles.Cen(cF-1,1) handles.Cen(cF-1,2)],...
          [handles.Cen(cF,1) handles.Cen(cF,2)]);
      end
      handles = loc_ps(handles,cF);
    end
    handles.pm(cF) = max(max(zT));
    if mod(cF,25)== 0
       tem = tem + 1;
       handles.ps(tem) = max(handles.pm(cF-24:cF));
       handles.dist(tem) = sum(handles.d(cF:-1:cF-24));
       handles.Adist(tem)= sum(handles.dist(1:tem));
       handles.vel(tem)  =  handles.dist(tem);
    end
end
handles.P = zT;
handles = getT(handles);
handles.output = hObject;
guidata(hObject, handles);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function varargout = GUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function start_Callback(hObject, eventdata, handles)
mov = aviread('H1-1.avi');
for i = 1:529
  x = handles.Bbox(i,1);          y = handles.Bbox(i,2);
  w = handles.Bbox(i,3);          h = handles.Bbox(i,4);
  x1 = ceil(handles.Cen(1:i,1));  y1 = ceil(handles.Cen(1:i,2));
  CF = mov(i).cdata;              BBox(CF,x,y,w,h,x1,y1,i,handles);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Trajectory_Callback(hObject, eventdata, handles)
handles.A = 1;    guidata(hObject, handles);
function PFD_Callback(hObject, eventdata, handles)
handles.A = 2;    guidata(hObject, handles);
```

```
function time1_Callback(hObject, eventdata, handles)
handles.A = 3;    guidata(hObject, handles);
function time2_Callback(hObject, eventdata, handles)
handles.A = 4;    guidata(hObject, handles);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function result1_Callback(hObject, eventdata, handles)
m = handles.A;   x = nonzeros(handles.Cen(:,1));
y = nonzeros(handles.Cen(:,2));     y = 240 - y;
z = find(handles.Cen(:,1));
if m == 1
 figure,plot3(x,y,z,'b-o')
 axis([0 320 0 240 0 530])
 xlabel('x')
 ylabel('y')
 zlabel('Frames')
 title('3D Plot For Human trajectory')
 grid on
 figure,plot(x,y,'b-o')
 grid on
 xlabel('x')
 ylabel('y')
 title('2D Plot For Human trajectory')
 figure,plot(z,x,'b-o')
 grid on
 xlabel('Frames')
 ylabel('X coordinates')
 title('X coordinate variation')
 figure,plot(z,y,'b-o')
 grid on
 xlabel('Frames')
 ylabel('Y coordinates')
 title('Y coordinate variation')
elseif m == 2
 figure,surf(double(handles.P(1:1:end,1:1:end))),zlim([0 handles.pm(end)]);
  axis([0 320 0 240 0 handles.pm(end)]);
elseif m == 3
    figure,bar(1:4,handles.T,0.3,'b');
    xlabel('Zone')
    ylabel('Time per sec')
    grid on
elseif m == 4
 figure,subplot(2,2,1)
 bar(1:4,handles.TA,0.3,'b')
 xlabel('Zone')
 ylabel('Time per sec')
 title('Time spend in zone 1')
 grid on
 subplot(2,2,2)
 bar(1:4,handles.TB,0.3,'b')
 xlabel('Zone')
 ylabel('Time per sec')
 title('Time spend in zone 2')
 grid on
 subplot(2,2,3)
 bar(1:4,handles.TC,0.3,'b')
 xlabel('Zone')
 ylabel('Time per sec')
 title('Time spend in zone 3')
 grid on
 subplot(2,2,4)
 bar(1:4,handles.TD,0.3,'b')
 xlabel('Zone')
 ylabel('Time per sec')
 title('Time spend in zone 4')
 grid on
else
 figure,plot3(x,y,z,'b-o')
 axis([0 320 0 240 0 530])
 xlabel('x')
 ylabel('y')
 zlabel('Time')
 title('3D Plot For Human trajectory')
 grid on
 figure,plot(x,y,'b-o')
 grid on
 xlabel('x')
 ylabel('y')
 title('2D Plot For Human trajectory')
 figure,plot(z,x,'b-o')
 grid on
 xlabel('Frames')
 ylabel('X coordinates')
 title('X coordinate variation')
 figure,plot(z,y,'b-o')
 grid on
 xlabel('Frames')
 ylabel('Y coordinates')
 title('Y coordinate variation')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function dist_Callback(hObject, eventdata, handles)
handles.U = 1;
guidata(hObject, handles);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function vel_Callback(hObject, eventdata, handles)
handles.U = 2;
guidata(hObject, handles);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%function
radiobutton13_Callback(hObject, eventdata, handles)
handles.U = 3;
guidata(hObject, handles);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function result2_Callback(hObject, eventdata, handles)
f = handles.U; tem = 0;
if f == 1
    figure,stem(handles.dist,'fill','b-')
    title('Distance Per Frame')
    xlabel('Numer of frames')
    ylabel('Crossed distance')
    grid on
    hold on
```

```
   plot(handles.dist,'-bo',...
            'LineWidth',3,...
            'MarkerEdgeColor','b',...
            'MarkerFaceColor',[0.01 0.01 1],...
            'MarkerSize',8)
   title('Distance Per Frame')
   figure,stem(handles.Adist,'fill','b-')
   hold on
   plot(handles.Adist,'-bo',...
            'LineWidth',3,...
            'MarkerEdgeColor','b',...
            'MarkerFaceColor',[0.01 0.01 1],...
            'MarkerSize',8)
   hold off
   title('Accumlated Distance Per second')
   grid on
elseif f == 2
   figure,plot(1:21,handles.vel,'-ro',...
            'LineWidth',2,...
            'MarkerEdgeColor','r',...
            'MarkerFaceColor',[1 0.01 0.01],...
            'MarkerSize',8)
   title('Object Velocity')
   xlabel('Time')
   ylabel('Velocity')
   grid on
elseif f == 3
 x = nonzeros(handles.Cen(1:8:end,1));
 y = nonzeros( handles.Cen(1:8:end,2));
 y = 240 - y;
 u = gradient(x);
 v = gradient(y);
 figure,quiver(x,y,u,v,'k'),
 xlabel('x'),ylabel('y'),
 title(' The Motion Direction')
 grid on
else
```

```
    figure,stem(handles.dist,'fill','b-')
    title('Distance Per Frame')
    xlabel('Numer of frames')
    ylabel('Crossed distance')
    grid on
    hold on
    plot(handles.dist,'-bo',...
                'LineWidth',3,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor',[0.01 0.01 1],...
                'MarkerSize',8)
    title('Distance Per Frame')
    figure,stem(handles.Adist,'fill','b-')
    hold on
    plot(handles.Adist,'-bo',...
                'LineWidth',3,...
                'MarkerEdgeColor','b',...
                'MarkerFaceColor',[0.01 0.01 1],...
                'MarkerSize',8)
    hold off
    xlabel('Time')
    ylabel('Image per pixel')
    title('Accumlated Distance Per second')
    grid on
end

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
s = stg(handles);
```