International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163
Issue 02, Volume 6 (February 2019) www.ijirae.com

DETECTION OF MOVING OBJECT

Ilapavaluri Umamaheshhwar Rao,
Scientist,PGAD/RCI,DRDO,Hyderabad, INDIA
iumrao@rediffmail.com;

Manuscript History
Number: **IJIRAE/RS/Vol.06/Issue02/FBAE10085**
Received: 17, January 2019
Final Correction: 05, February 2019
Final Accepted: 26, February 2019
Published: **February 2019**
Citation: Rao, U. (2019). DETECTION OF MOVING OBJECT. IJIRAE::International Journal of Innovative Research in Advanced Engineering, Volume VI, 45-49. **doi://10.26562/IJIRAE.2019.FBAE10085**
Editor: Dr.A.Arul L.S, Chief Editor, IJIRAE, AM Publications, India

Abstract: Surveillance refers to the task of observing a scene, often for lengthy periods in search of particular objects or particular behaviour. This task has many applications, foremost among them is security (monitoring for undesirable behaviour such as theft or vandalism), but increasing numbers of others in areas such as agriculture also exist. Historically, closed circuit TV (CCTV) surveillance has been mundane and labour Intensive, involving personnel scanning multiple screens, but the advent of reasonably priced fast hardware means that automatic surveillance is becoming a realistic task to attempt in real time. Several attempts at this are underway.

Indexing terms: helix; moving coil; surveillance; detection;

I.INTRODUCTION: PASSIVE SURVEILLANCE

For simplicity, consider a scene being monitored by a stationary camera Given A Sequence of such scenes ,we might hope that any significant change between them may be due to objects of interest in motion-in this example pedestrians, But other applications may pick out vehicles ,animals or some other object. In reality, changes in weather conditions, slight movement of objects such as Trees, camera judder etc. can cause there to be many changes from frame to frame it is possible ,however , to maintain a good background estimate by computing the median of the last few frames(perhaps over the last 5 minutes). Mcfarlane gives an efficient way of doing this without storing all the relevant information I Computing image differences frame by frame with such a dynamically maintained reference provides a no of 'blobs' which when threshold for size, usually provide a Silhouette of the objects of interest, except that the difference image obtains the absolute intensity difference(i.e. is not binary). This difference is blurred and then threshold to reduce i.e. effects of noise. Various quantization and remaining= noise effects leave bound. Simple perfect, but a simple sequence of morphological operations usually enough to produce usable shapes, a sequence of dilations Anderosions is sufficient that may appear in shapes. A lengthy Sequence of input will generate a large no of such silhcuettes which we may Then attempt to analyze for shape characteristics 4'hc ap (n oachis to approximate The boundary of the region with a cubic B—spline First a star (hard refitence point On the boundary is needed; for this application,this isr Jiinc straiuhiii›rward Dy determining the principal axis of the shape. For sténes involving humans in upright positions, this usually works well in identifying the top of the around the boundary are used to derive a B-spline representation with 40 controlled points. Given a training set of such shapes, it is now straight forwardto analyze them Using point distribution model method. Each shape has 80 parameters(from 40 control points)(x1,yl,x2,y2,... ,x40,y40),but it transpires that the first 18 eigen values account for virtually all the variation detected. Thus we are representing a shape as a vector b,

$$X=\bar{x}+Pb \quad \text{......Eq.1.1}$$

Where X is the 80—dimensional vector defining the spline. Xis the mean shape, P is An80, matrix, a and b is an 18-dimensional vector parameterizing the shape X. The first moae captures the difference between silhoucttees in which one and two legs are visible.

The second clearly indicates swaying of a moving figure. An immediate application of this analysis is to clean errors in siJhouettes;ive can take a noise boundary X ,determine the PDM boundary b,project this into the closest point b within the model space defined by a (clean) training set, and map this back to a spline defined by a vector x which is normally cleaner than the original. Strictly, w that we are doing is constraining how far the eigenco-ordinates may vary, and requiring that b lie in an 18D hyper-ellipsoid. A primary aim of this modelling is to assist in tracking that is to detect an object (pedestrian) and follow its trajectory through the scene. Given the PDM representation we note that a spline appears in the scene after due trans1ation,rotation and scaling; if the current offset is (ox; oy),the scale is s and the rotation 8 ,we may model the boundary by

Q=(s cosø -s sinø;ssinø s cosø)…….1.2

where xiyi aregiven by eqI. 1. Then if we write o=(ox,oy,ox,oy,...ox,oy)

(40times) and Q=(Q…….O,……O Q)……1.3

(an80* 80matrix) t hen the shape vector x is related to the state b by the equation

X=Q(Pb+x¯)+o…….1.4

When a new object is detected, it will not be clear what the best estimates of its scale, trajectory, or model parameters are ,but given suitable assumptions we i right initialize these parameters and then iterate them during successive frames using a kalman filter to converge on a goocos d estimate. We suppose that the object to be initialized has bounding box given by lower left co-ordinates (xl,y1) and upper right (xr, yr), and that the mean height of figures in a training set is hm. Rewriting equation 1.2as

(scosø -s sinø ; ssinø s cosø) = (ax -ay; ayax)

We can then initialize t1ie figure as the mean shape b°=0 and ax°= (yr-y1)/hmay°=0 ox°=(xl+xr)/2 oy°=yl+yr)/2

So the figure is scaled to its bounding box, aligned vertical1y, with the origin at the center of the box. A reasonable model for the evolution of the.se parameters is to assume that Nie object is moving uniformly in 2D subject to additive noise. Roe frame update equation

(ox°^(k+ 1);ox°^(k+1)=(1delta t 0 delta t)(ox^k;ox-°^k)+(vx;wx) 1.5

The kalman filter proceeds by estimating origin, alignment and shape Independently of another given estimates of the parameters, we predict the next state. X^- from equation 1 4 and use this to make an observation from the Image z. To› update the x-origin co—ordinate we need to consider the state (ox^, ox. °^) Many measurements of the origin are available from the expression. Z=Q(Pb^+x¯. These measurements and eq 1...5 and the noise variance properties are used to provide the best estimate of the origin in the next frame.,

OSCILLATIONS OF A MOVINO COIL-HELIX

Let us Suppose that we want to visualize the oscillations of a simple moving Coil spring whose shape is that of a helix. The parametric equations of this curve are X=d/2 COS P Y=d/2 SIN P. Where d is the diameter of the helix. To similar oscillations with Amplitude 'A' and angular frequency o we modify the equation of z to Z=(1+A COSɯ(t-1)))P - Where t is the time variable. We begin by plotting a reference trame, for example for t=l and check the Appearance of the plot on it .P= 0: pi/60:8*pi
D=2;A=0.2;T=5;
Omega=2*pi/T;
%é pIot reference frame
x=d*cos(p)/2; y=d*sin(p)/2;
z=(I+A*cos (omega*1))*p;
plot3(x,y,z)
The next step is the generation of a number of frames say six. We begin by timing axes that will be sufficient for all frames. The statement Movie in (6) Creates a matrix with 6 columns one for each frame. The frames themselves are CieneratedwithinaFORloop.Thegetfraniefunctionreturnsapixelimageoftlietofoscillating HELIX and SHAPE vector are included in this document and also the MATLAB program

% APPENDIX-A -MATLAB PROGRAM TO DETECT MOVING OBJECT EXAMPLE MOVING COIL OSCILLATING

```
p =0:pi/60:8*pi;
d=2;A=0.2;T=5;
omega=2*pi/T;
%PLOT REFERENCE FRAME
x=d*cos(p)/2;
y=d*sin(p)/2;
z=(1+A*cos(omega*1))*p;
%figure
```



```
%plot3(x,y,z)
%6 frames matrix of  6 columns one each  for frame
% the get frame function retuns a pixel image of the current figure
M=moviein(6);
for t=1:50    %record the movie
   x=d*cos(p)/2;y=d*sin(p)/2;
   z=(1+A*cos(omega*(t-1)))*p;
   %figure
   plot3(x,y,z)
axis([-1 1 -1 1 0 10.0*pi]);
M(:,t)=getframe;
end
%load('helix.jpg');
%[X,Map]=IMREAD('helix','jpg');
X=[x,y,z];
%step1: access an image acquisition device
% %access an image acquisition device
% vidobj=image(X);
% %configure the no.of frames to log
% vidobj.FramesPerTrigger=50;
% %skip the first few frames the device provides before logging data
% vidobj.TriggerFrameDelay=5;
% %access the device’s video source
% src=getselectedsource(vidobj);
% %configure the device's  frame rate(frames per second)
% actualRate=15.15;
% src.FrameRate=num2str(actualRate);
% %step2:log and retrieve data
% %start the acqisition
%start(vidobj)
%wait for data logging to and before retrieving data
%wait(vidobj,10);
%retrieve the  data and timestamps
%[frames,timestamp]=getdata(vidobj);
xx=X(1:10.5:420);
yy=X(1:14:560);
xxx=[xx;yy];
xbar=mean(X);
for i=1:18
b(i)=X(i);
p(i)=b(i).*xxx(i);
end
for i=1:18
PB(i)= p(i).*b(i);
end
lb=length(b);
lp=length(p);
lpb=length(PB);
%PBXBAR= (PB)+xbar(1,lpb);
%plpxbar=length(PBXBAR);
%CURRENT OFFSET IS (ox,oy);
%scale
s=10;
%rotation
theta=90;
Q=[s*cos(theta)    -s*sin(theta);s*sin(theta)    s*cos(theta)];
```



```
ox=2;oy=3;
lq=length(Q);
QS=Q.*255;
XX=QS+((2)+(3));
plot3(XX,XX,XX);
xl=5:2:95;yl=3:3:55;%LOWER LEFT CO-ORDINARTES OF IMAGE BOX
xr=xl;yr=yl;%UPPER RIGHT CO-ORDINATES OF IMAGE BOX
hm=90;%MEAN HEIGHT OF IMAGE]
ayo=0;oxo=(xl+xr)./2;
oyo=(yl+yr)./2;
q=20;
r=20;
t=[0:100];
randn('seed',0)
w=sqrt(q)+randn(length(t),1);
v=sqrt(r)*randn(length(t),1);
dox=diff(oxo);
k=0.05;
%FRAME UPDATE EQUATION
for i=1:45
deltat(i)=2;
oxox(i)=oxo(i);
vv(i)=v(i);
ww(i)=w(i);
end
%-------------------------------------------------------------------
%FRAME UP DATE EQUATION  USING KALMAN FILTERING METHOD
%--------------------------------------------------------
Z=(oxox.^(k+1));(dox.^(k+1))==(deltat.^k).*(oxox.^k).*(dox.^k)+(vv)+(ww);
%Z IS ESTIMATED VALUE
m=image(X);
%XXM=QS.*255;
%for i=1:16
org=m-XX;
%end
plot(org);
%plot(org)
plot(Z);
```

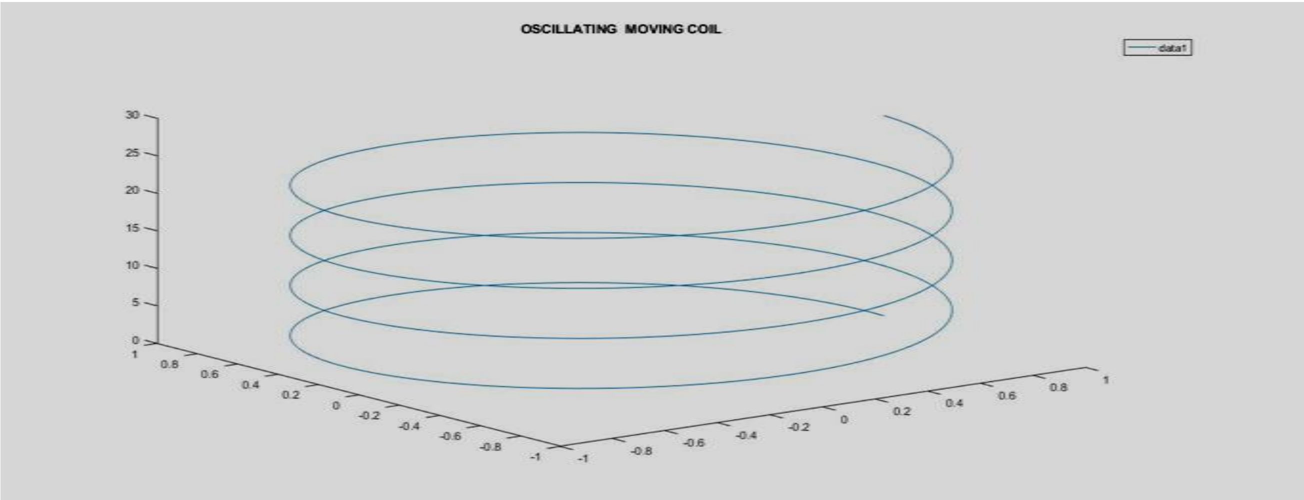

FIGURE1: AN OSCILLATING MOVING COIL: HELIX

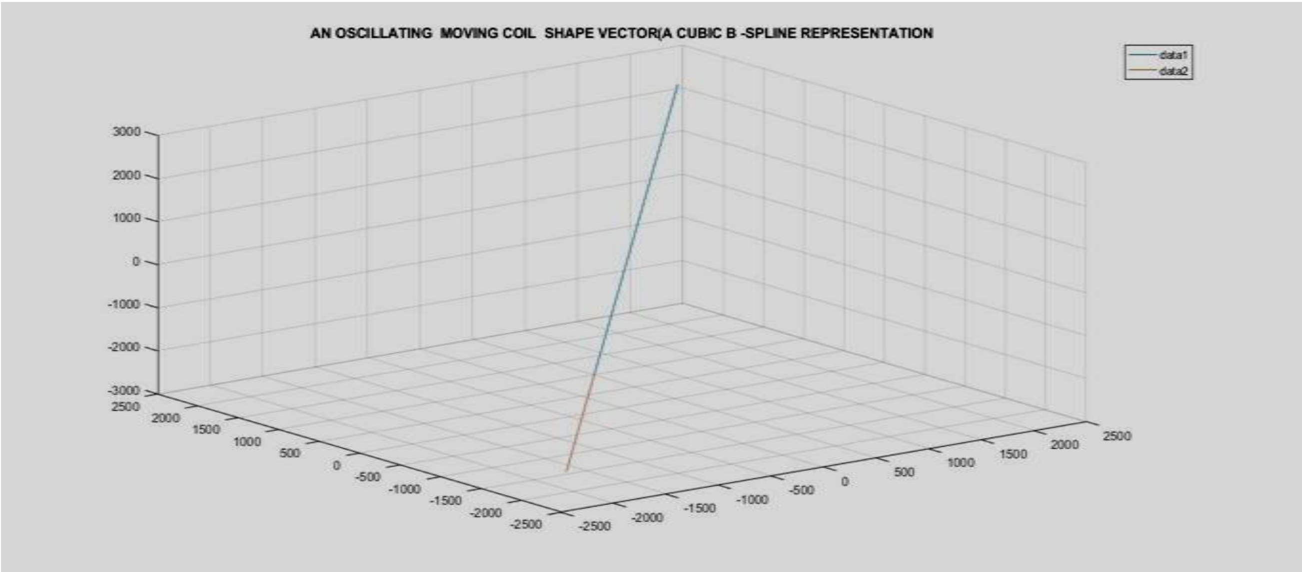

FIGURE2: SHAPE VECTOR OF MOVING COIL

FIGURE3:2 PERSONS MOVING (WALKING PERSONS) JUST LIKE MOVING OBJECTS