EE 417 POSTLAB

REPORT #6

Name:Nidanur Gunay

ID:24231

Task 1 : Optical Flow

Basically optical flow is about understanding how things are moving in an image, how motion is happening in pixel level. This understanding helps us for camera stabilization. So this is about seeing how the global motion and all global surfaces in the scene are moving around. In addition, the object that is closer to camera moves faster. By the help of this information, you can have a depth perception.

.

Things to do:

In this lab, you will write a program ("lab6OFMain.m") and a function ("lab6OF.m") to calculate the optical flow throughout a video.

Optical flow is the distribution of the apparent velocities of objects in an image. By estimating optical flow between video frames, one can measure the velocities of objects in the video. These velocities can be estimated by:

$$G = \begin{bmatrix} \sum_{p \in W} I_x^2 & \sum_{p \in W} I_x I_y \\ \sum_{p \in W} I_x I_y & \sum_{p \in W} I_y^2 \end{bmatrix} \quad b = \begin{bmatrix} \sum_{p \in W} I_x I_t \\ \sum_{p \in W} I_y I_t \end{bmatrix}$$

$$u = [Vx; Vy] = -G^{-1} b$$

where; I_x and I_y are spatial gradients in X and Y directions, I_t is the gradient in temporal direction, V_x and V_y are velocity vectors.

Your resulting images will look as follows:

RESULTS:

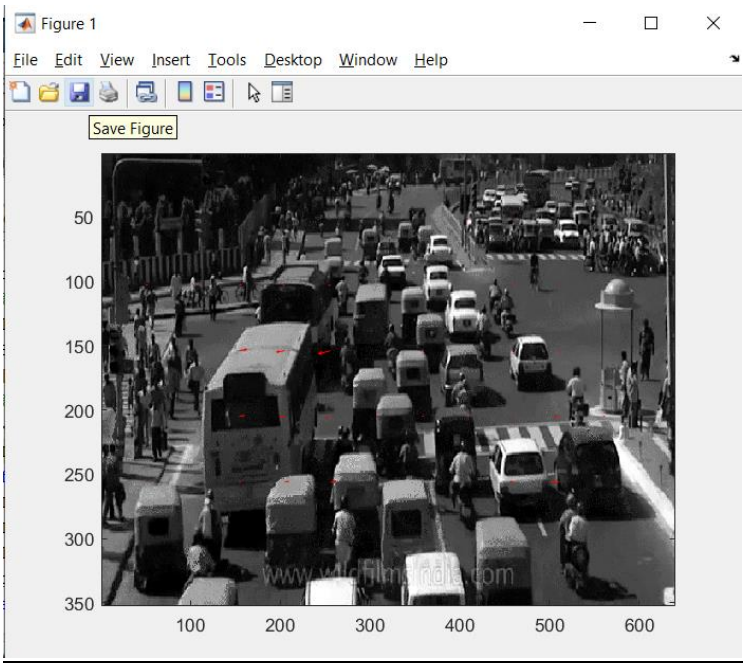
TASK 2: K VALUE COMPARISON

Firstly, I've tried to set k value as 10, 20 and 30 as how it described in PostLab pdf. However, I've encountered error that k value needed to be odd number thus I set the k value as an odd number. In order to observe that change in k value how affects the optical flow, I changed the values and observed the results.

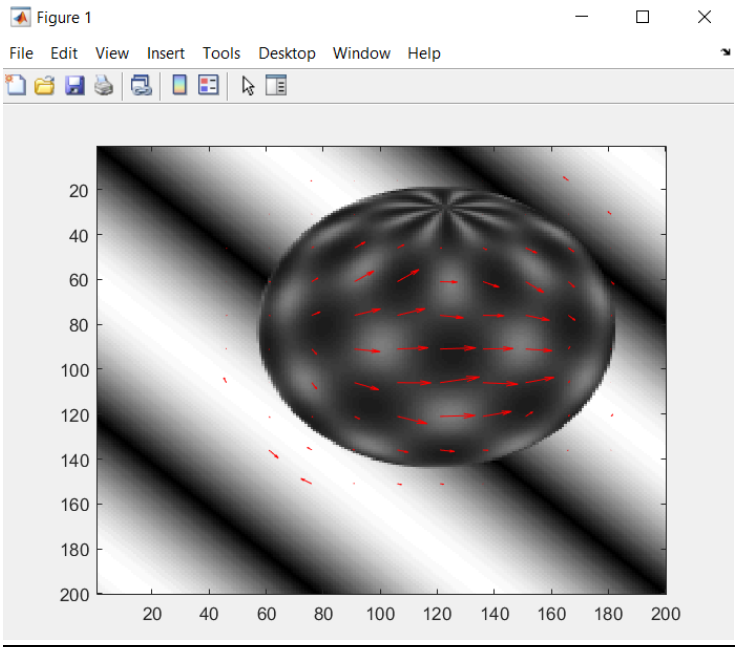
OBSERVATION OF K VALUE & RESULTS:

Seq: Cars2 K=15 T=3000

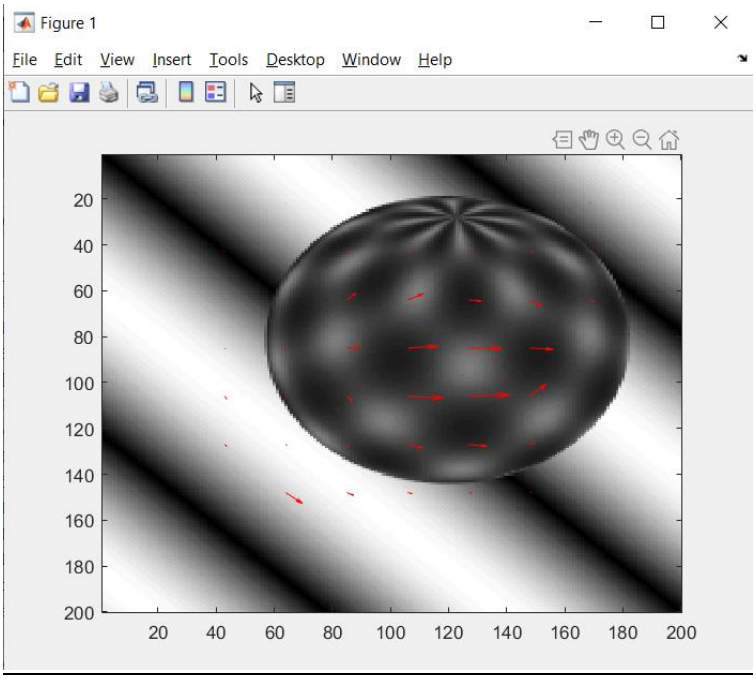
Seq: Cars2 K=51 T=3000



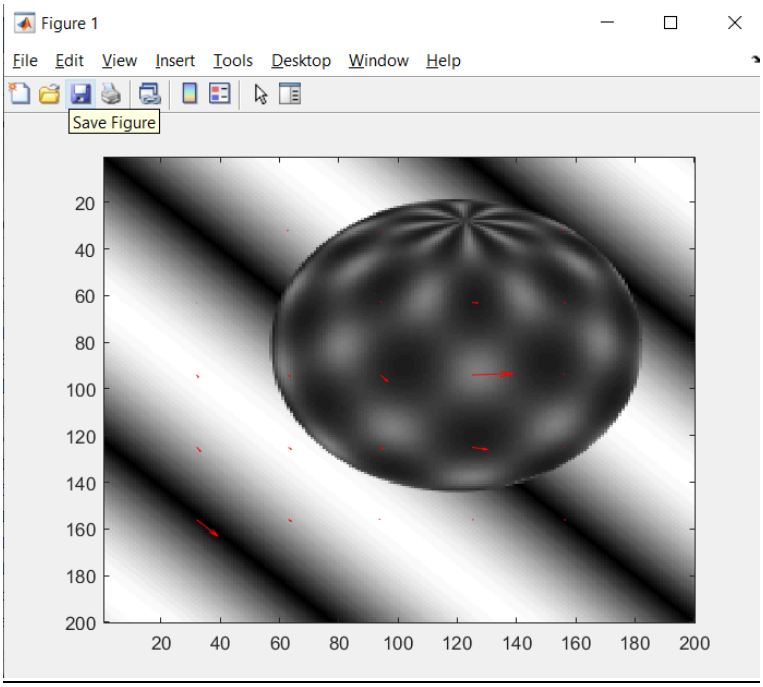
Seq: Sephere K=15 T=3000



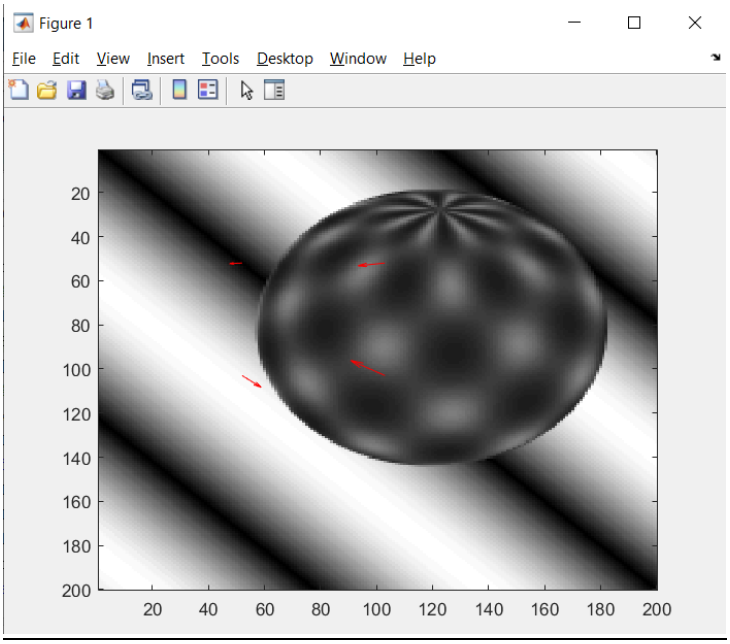
Seq:Sphere K=21 T=3000



Seq:Sphere K=31 T=3000



Seq:Sphere K=51 T=3000

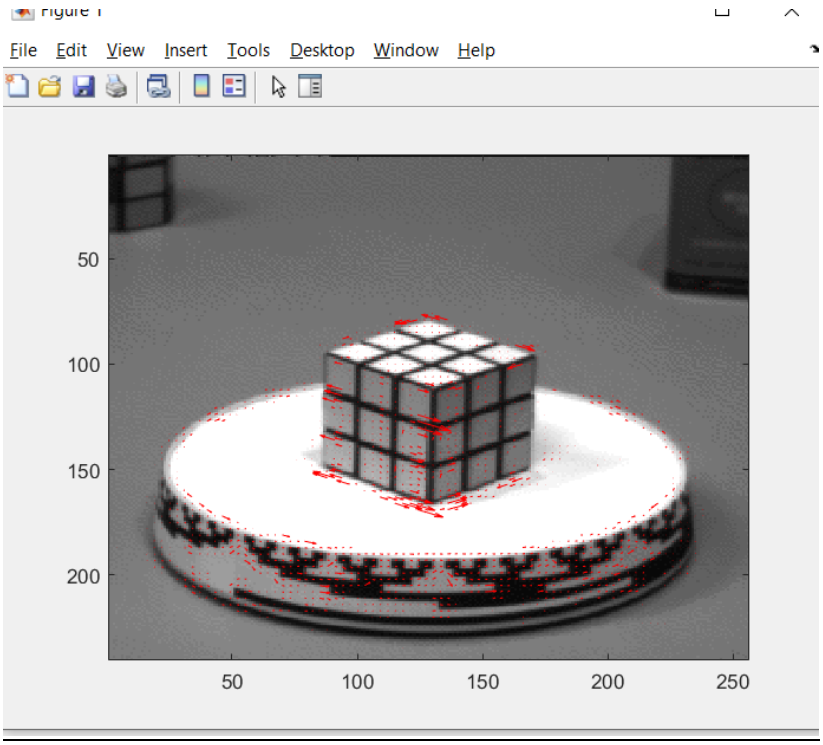


Seq: Taxi K=3 T=3000

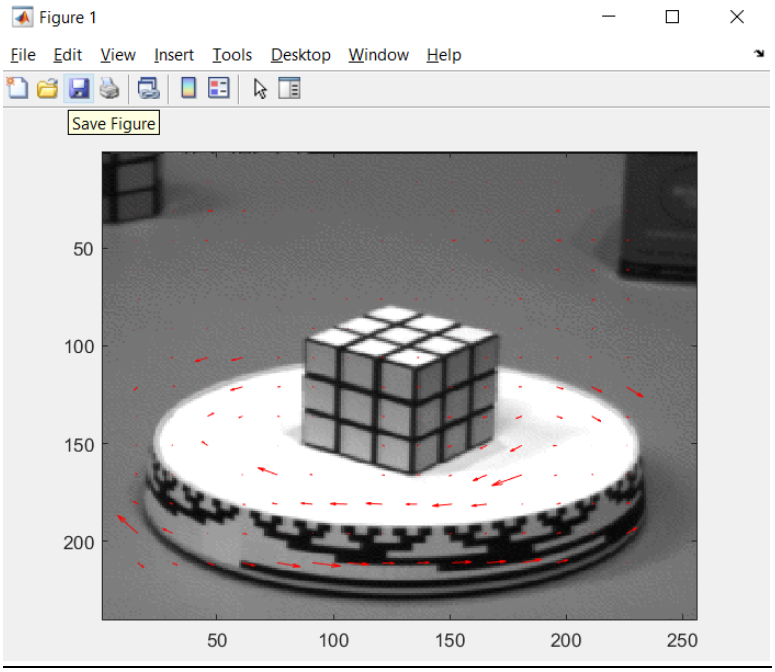
Seq:Taxi K=15 T=3000

Seq:Taxi K=31 T=3000

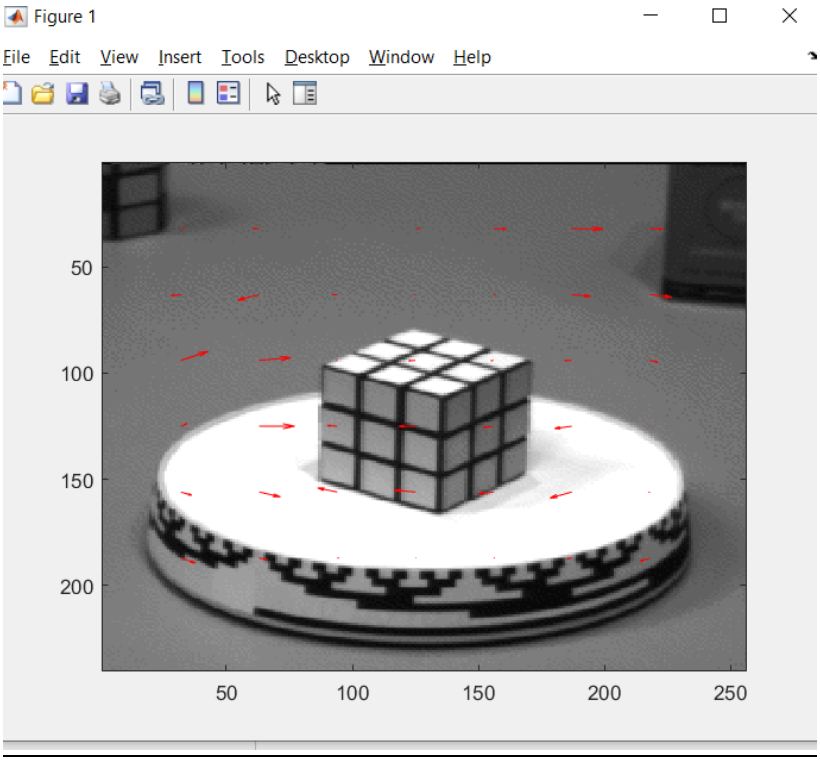
Seq:Rubic K=3 T=3000



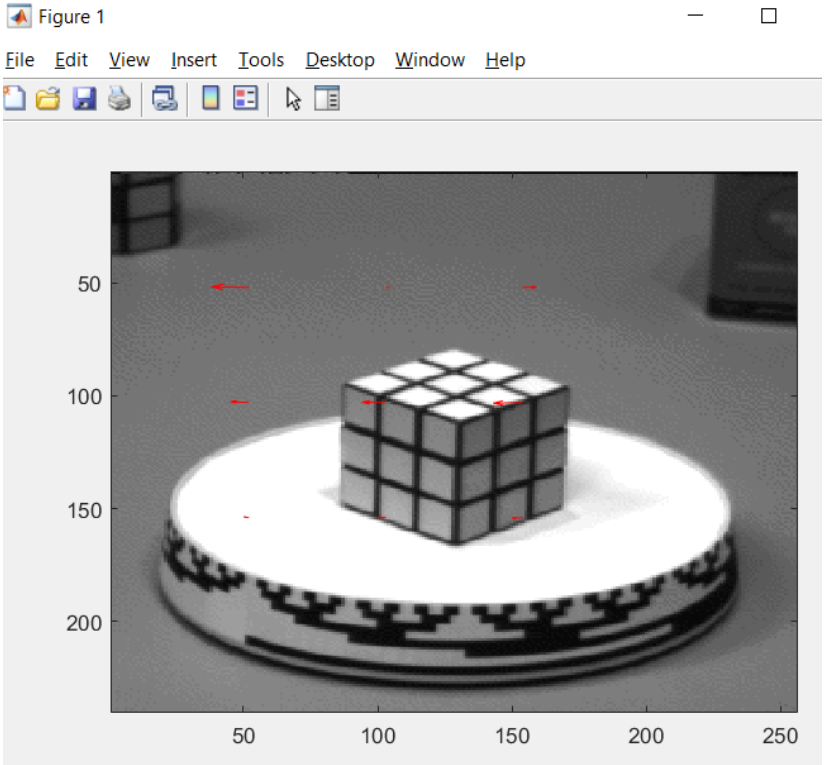
Seq:Rubic K=15 T=3000



Seq:Rubic K=31 T=3000



Seq:Rubic K=51 T=3000

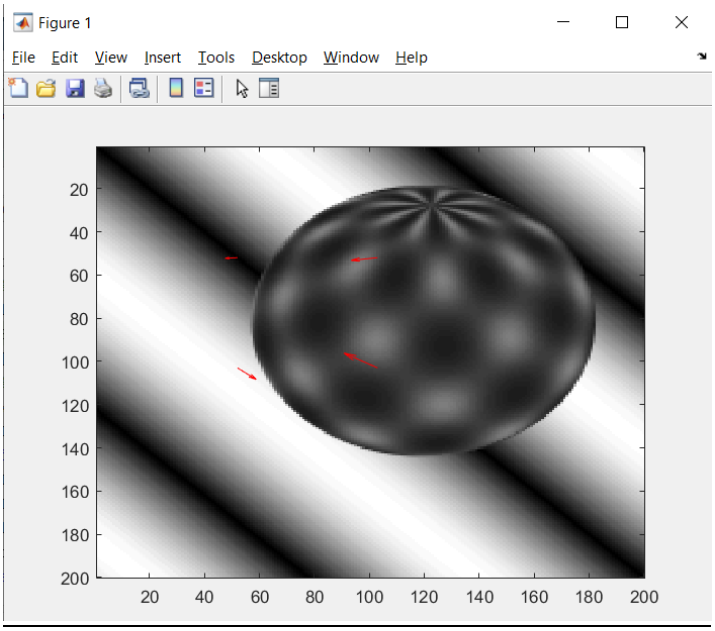


DISCUSSION:

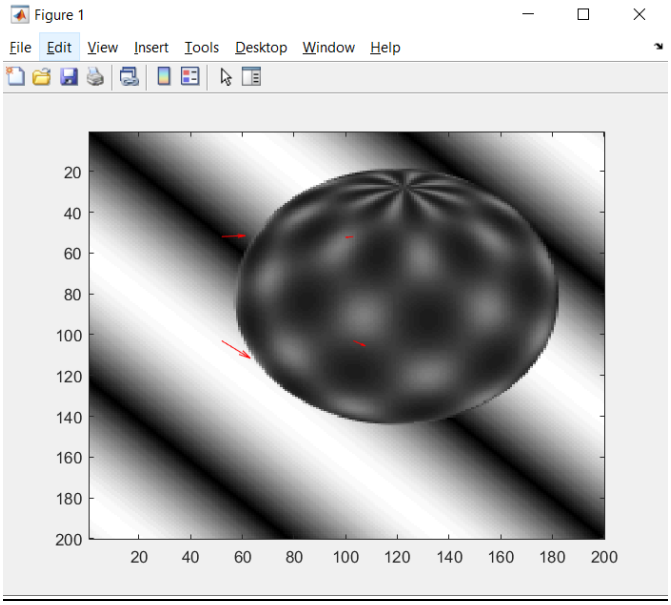
I changed the k value in various video sequences. In each sequence, when we increase the k value, number of arrows decrease. It happened because k value also determines the window size. When we set the k value smaller, since the window size is also getting smaller, we can observe the motion in almost pixel level and it gives us more accurate solution. When we pick k as a large number distance between arrows are also getting larger and arrows become more separate to each other. Thus, we should set k value as small as we can in order to track of the object's motions more accurately.

OBSERVATION OF FILTER EFFECT & RESULTS

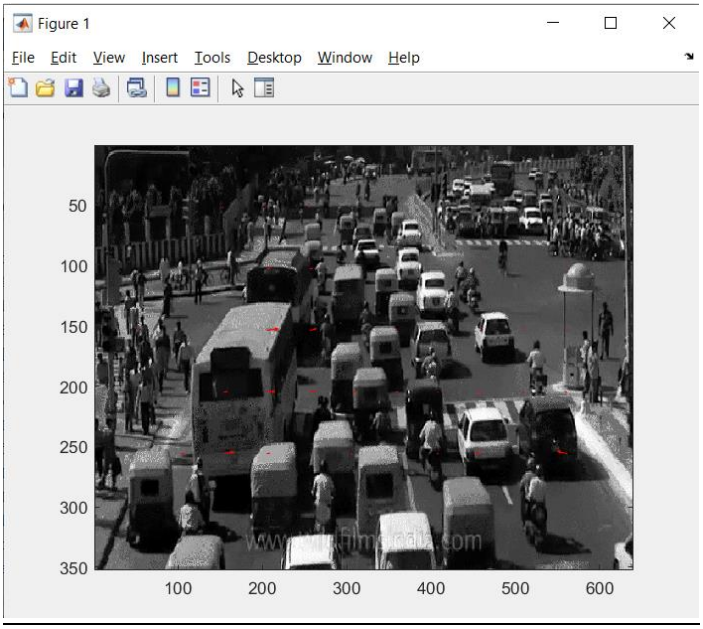
Seq:Sphere K=51 T=1000 Box Filtered



Seq:Sphere K=51 T=1000 Gaussian Filtered

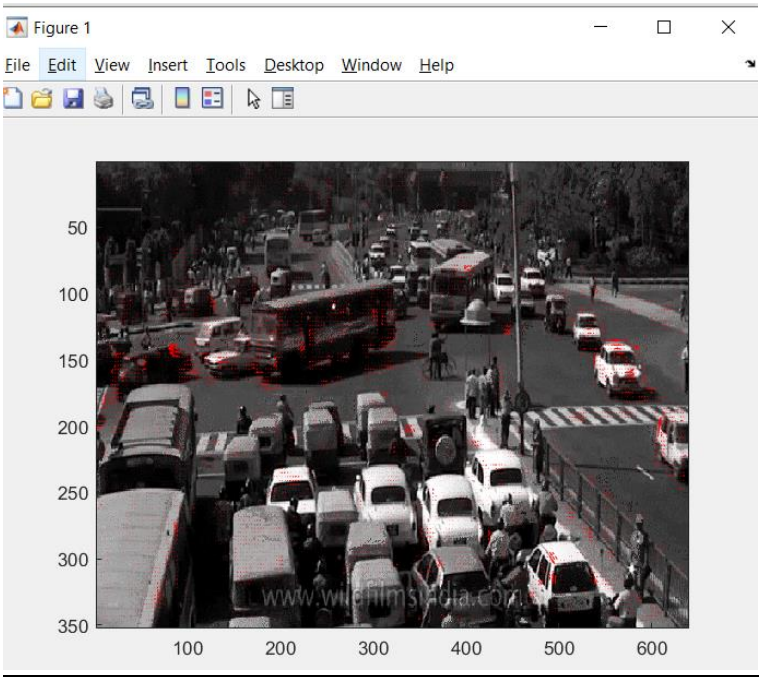


Seq:Cars2 K=51 T=1000 Box Filtered

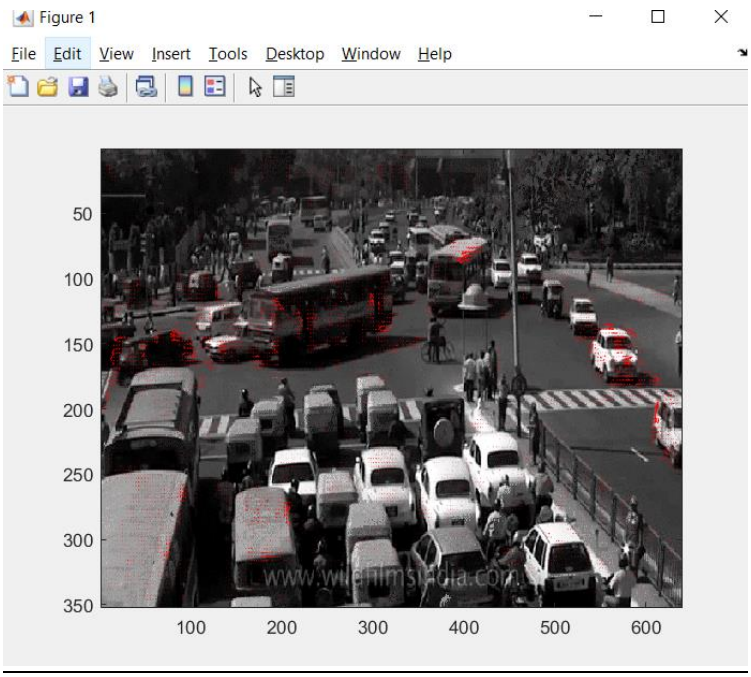


Seq=Cars2 K=51 T=1000 Gaussian Filtered

Seq:Cars1 K=3 T=3000 Box Filtered

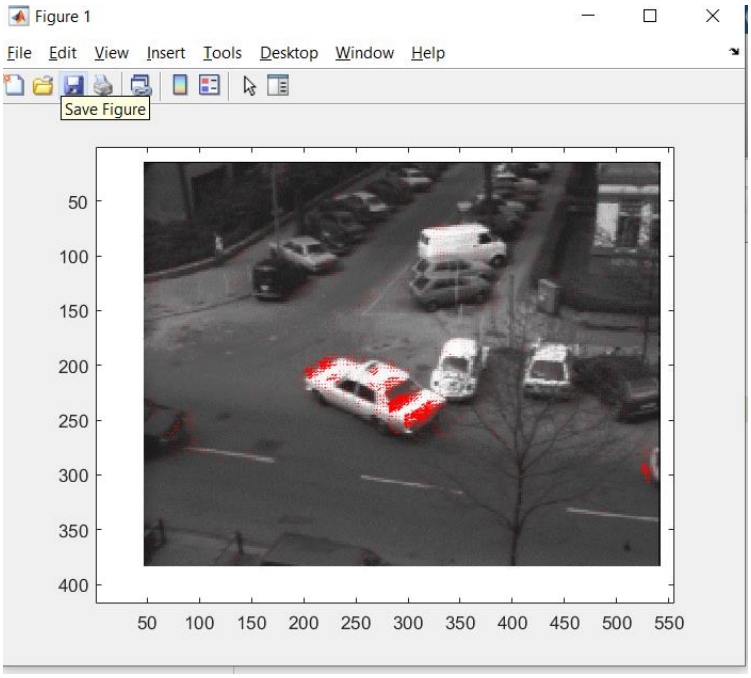


Seq:Cars1 K=3 T=3000 Gaussian Filtered

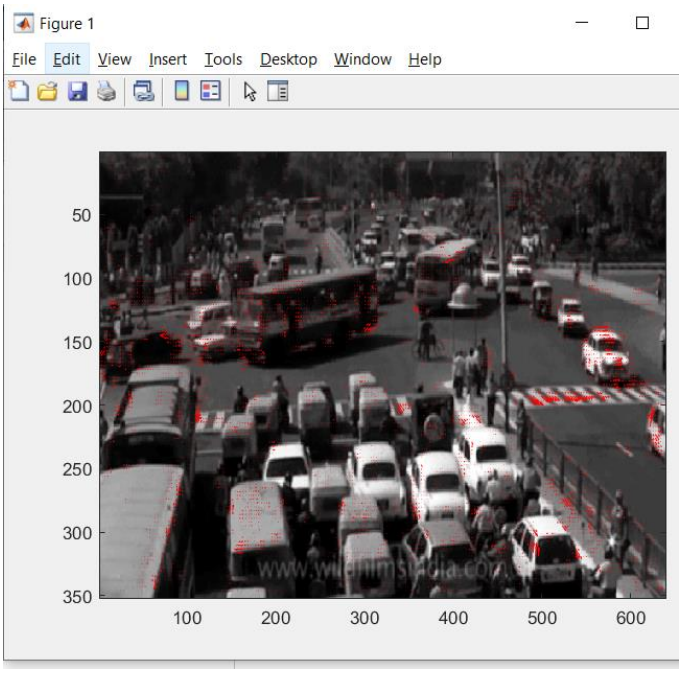


Seq:Taxi K=3 T=3000 Box Filtered

Seq:Taxi K=3 T=3000 Gaussian Filtered



Seq:Cars1 K=3 T=3000 Box Filtered



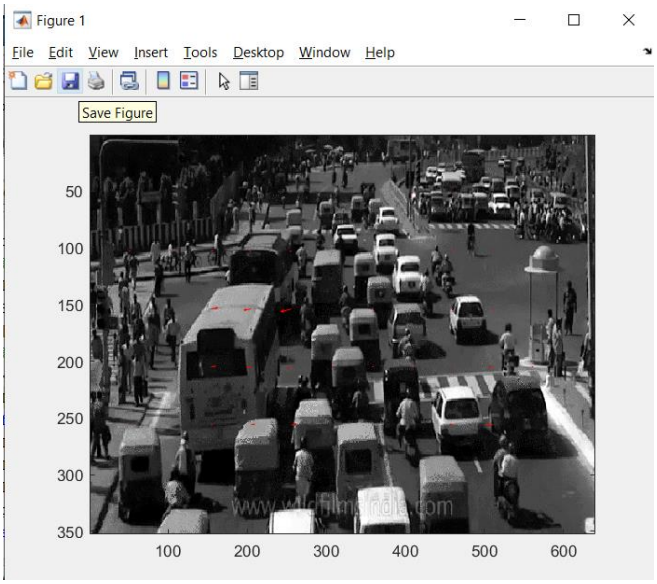
Seq:Cars1 K=3 T=3000 Gaussian Filtered

DISCUSSION:

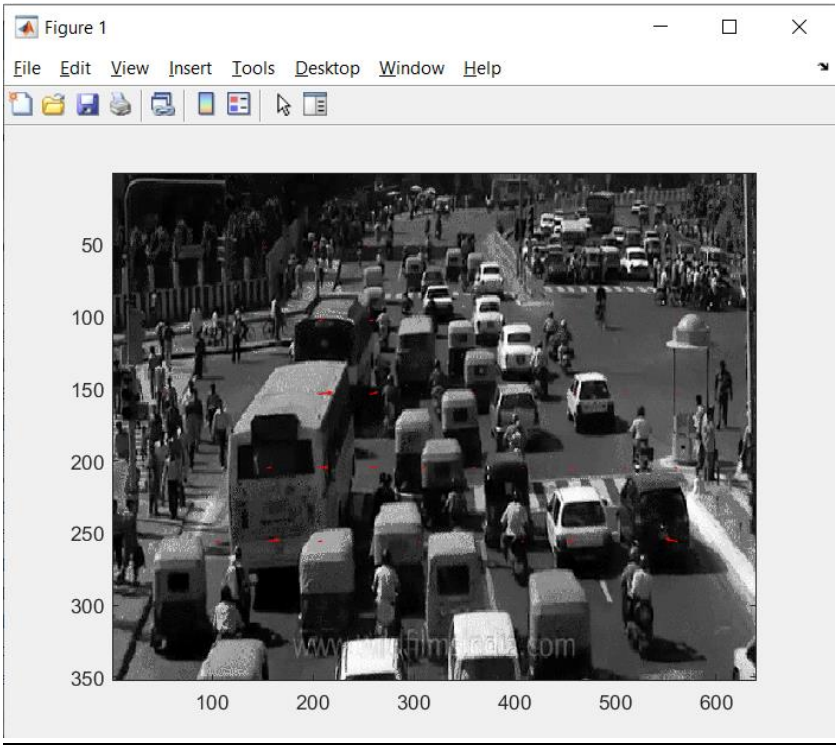
In the first 2 sequences it as hard to make an observation about the effect of filter due to large k value. When we set window size as 3, we can easily observe how filter affects the optic flow. When we use gaussian filter we can see that motion vector arrows are denser. However, in Box filter they are separate to each other. In last 2 screenshots are taken for the observing Box and Gaussian Filter in main video frame sequences. Since Gaussian filter smoothing frames more than the Box filter, density of motion vectors increases and some of them disappear. In Cars1 video frame sequences, the pedestrian crossing does not move however when we use box filter, motion vectors in pedestrian crossing appear. In despite of this in gaussian filtered video frames it does not appear. As a conclusion, In order to detetctig moving objects, gaussian filter gives more accurate solution.

OBSERVATION OF THRESHOLD & RESULTS

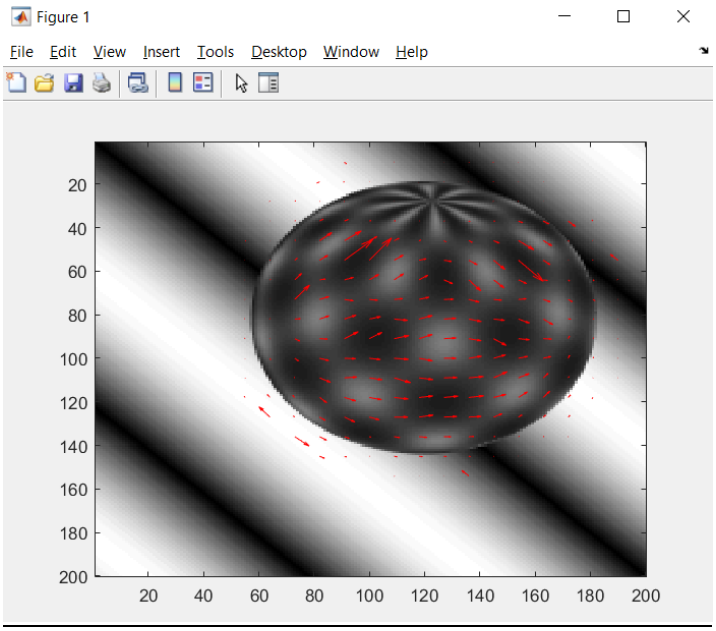
Seq:Cars2 K=51 T=1000



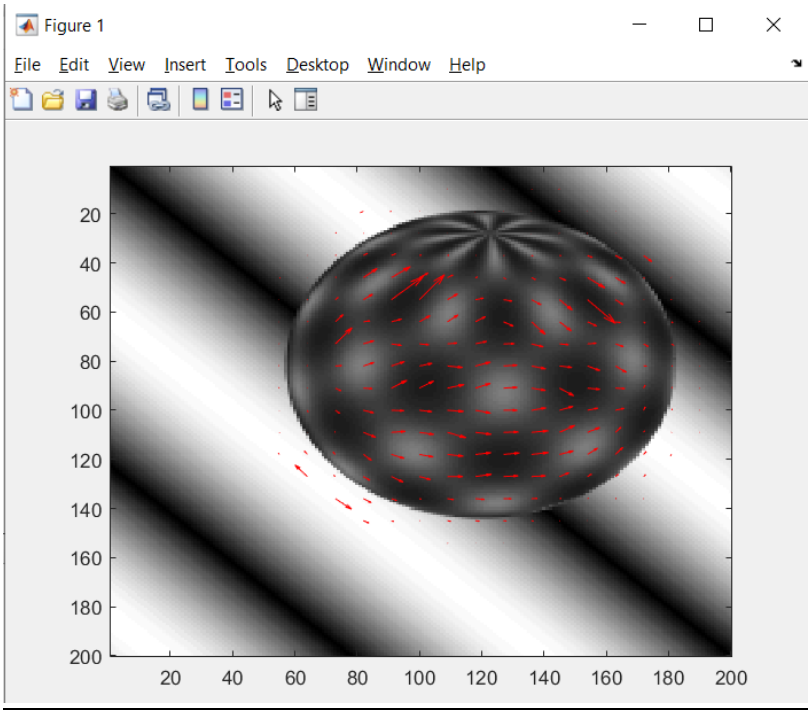
Seq:Cars2 K=51 T=5000



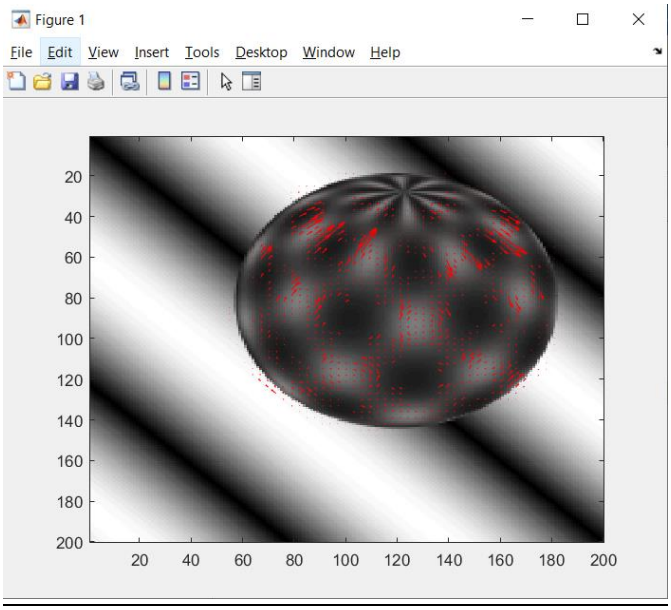
Seq:Sphere K=9 T=500



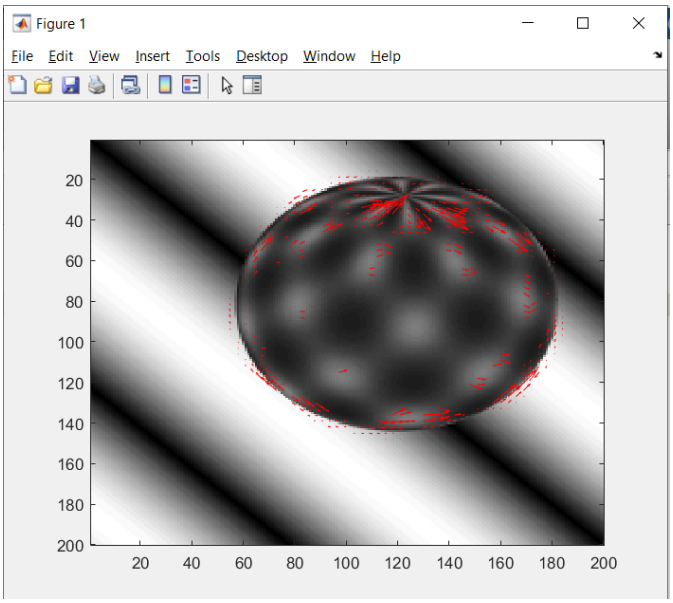
Seq=Sphere K=9 T=3000



Seq:Sphere K=3 T=1000



Seq:Sphere K=3 T=10000



DISCUSSION:

When I get first two image sequences, I conclude the outputs as threshold value does not affect the output. However, when I set the k value smaller the difference becomes apparent. Since, smaller motion vectors could not reach the high threshold value, they can not appear. Last 2 frames screenshots support my hypothesis. Motion vectors that is located in the center disappears since the motion difference is low in the central area of hemisphere.

APPENDIX:

lab6OFMAIN.m

```
clear all; close all; clc;

% Load the files given in SUcourse as Seq variable
load('traffic.mat');
Seq=traffic;
[row,col,num]=size(Seq);
% Define k and Threshold
k=15;
Threshold=3000;
for j=2:1:num
ImPrev = Seq(:,:,j-1);
ImCurr = Seq(:,:,j);
lab6OF(ImPrev,ImCurr,k,Threshold);
pause(0.1);
end
```

lab6OF.m

```
function lab6OF(ImPrev,ImCurr,k,Threshold)
% Smooth the input images using a Box filter
ImPrev=double(ImPrev);
ImCurr=double(ImCurr);
ImPrev1=imboxfilt(ImPrev,k);
ImCurr1=imboxfilt(ImCurr,k);

% Calculate spatial gradients (Ix, Iy) using Prewitt filter

[Ix,Iy]=imgradientxy(ImCurr,'Prewitt');

% Calculate temporal (It) gradient
It=-ImPrev1+ImCurr1;
[ydim,xdim] = size(ImCurr);
Vx = zeros(ydim,xdim);
Vy = zeros(ydim,xdim);

G = zeros(2,2);
b = zeros(2,1);
cx=k+1;
for x=k+1:k:xdim-k-1
cy=k+1;
    for y=k+1:k:ydim-k-1
% Calculate the elements of G and b

        G(1,1) = sum(sum(Ix(y-k:y+k, x-k:x+k).^2));
        G(1,2) = sum(sum(Ix(y-k:y+k, x-k:x+k).*Iy(y-k:y+k, x-k:x+k)));
        G(2,1) = sum(sum(Ix(y-k:y+k, x-k:x+k).*Iy(y-k:y+k, x-k:x+k)));
        G(2,2) = sum(sum(Iy(y-k:y+k, x-k:x+k).^2));
        b(1,1) = sum(sum(Ix(y-k:y+k, x-k:x+k).*It(y-k:y+k, x-k:x+k)));
```

```
        b(1,2) = sum(sum(Iy(y-k:y+k, x-k:x+k).*It(y-k:y+k, x-k:x+k)));


        e=eig(G);
        if (min(e) < Threshold)
        Vx(cy,cx)=0;
        Vy(cy,cx)=0;
        else
        % Calculate u
        Ginv=pinv(G);
        u=-(Ginv*b);

        Vx(cy,cx)=u(1);
        Vy(cy,cx)=u(2);
        end
    cy=cy+k;
    end
    cx=cx+k;
end

ImPrev=uint8(ImPrev);
ImCurr=uint8(ImCurr);
cla reset;
imagesc(ImPrev); hold on;
[xramp,yramp] = meshgrid(1:1:xdim,1:1:ydim);
quiver(xramp,yramp,Vx,Vy,10,'r');
colormap gray;
end
```