Computer Vision LAB#4

Report Submitted By Ahmed, Muhammad Farhan , EMARO

**LUCAS KANADE Method for Optical Flow**

The Lucas Kanade method, also known as sparse optical flow, calculates the displacement vectors of individual features rather than tracking all of the pixels within a frame and rendering a full motion vector field. This method uses gradient information obtained from consecutive frames to search along the optimal path for a region that best matches the desired feature.

In order to track an object, we first identify well-textured features within the target region and then calculate their corresponding optical flow in each new image using a two-frame gradient-based method developed by Lucas and Kanade. The task of matching a region in the current frame ‘t’ to a target region in the past frame t-1 is generally referred to as a registration problem. Optical flow is a registration method that provides a measure of the apparent motion within a sequence of images.

For each position **x**=(x,y) of the image, we consider a NxN neighborhood (e.g. N=5, 7) Q

which produces a system of NxN equations in the two unknowns (ux, vy) =**u** (which are the

components of the optical flow computed in position **x**). If we denote with **x**i the points in Q,

Then



The left side of the equations is based on the computation of the image gradients (see the lab.

session on edge detection), while on the right of the equal there is the temporal derivative,

which may be approximated as

Now, the NxN equations form a system of type A**u**=**b,** that can be solved as



**Procedure/steps in main program**

1. Load the images form a specific location to into a structure, then read images at 2 images at a time i.e. image at t and image at t+1. Pass these images to for function MyLucasKanade and obtain matrices for Ux and Vy. (we take initially 10 images for each type).
2. Plot Ux and Vy using quiver, scale images if necessary.
3. By default the main displays optic flow for sphere only but user can uncomment corresponding code for other image types.
4. The function MyLucasKanade operates in following manner
5. It receives three input arguments image1 ,image2 and N(size of Neighborhood).
6. We smooth each image by Gaussian kernel of suitable size e.g. 5X5 and S.D = 0.1. To suppress noise components.
7. Calculate the derivative along x-axis and y-axis for both images using kernel [1,-1] and [1,-1]’
8. Calculate NXN neighborhood for each pixel in both images Qt and Qt+1 .
9. Find difference I(Xi,t).
10. Calculate matrix A by adding dx and dy matrices of image1(at time instant t).
11. Calculate U by applying pseudo inverse u=A’\b;
12. Separate matrices for Ux and Vy form U.
13. Return Ux and Vy to main program to display them.

**Results**

**FOR Sphere**

Figure No.1 Figure No.2

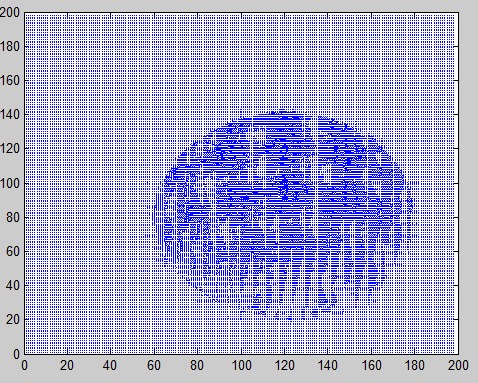
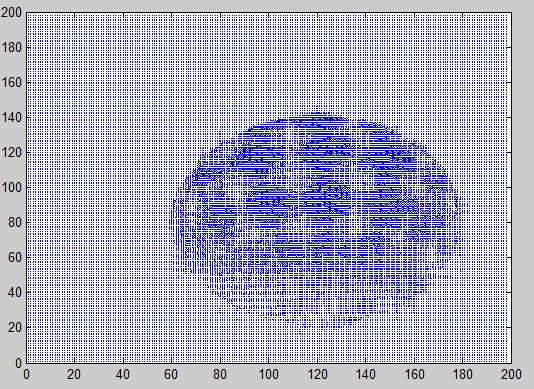
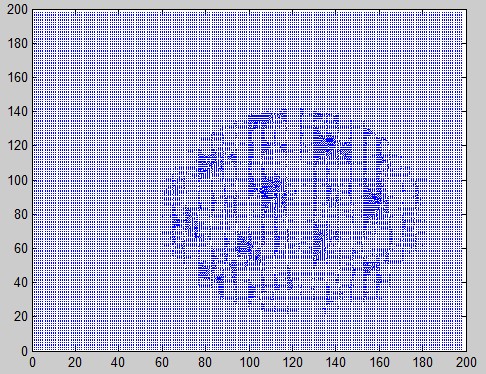
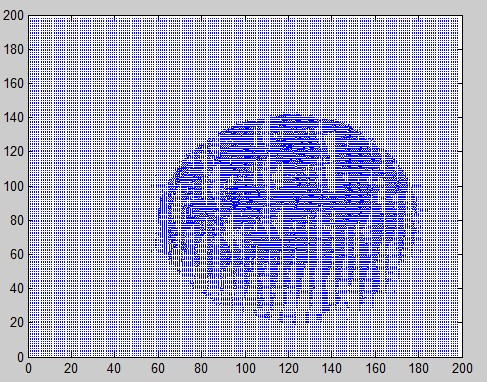


Figure No.3 Figure No.4



**FOR Tree-Sunflower**

Figure No.1 Figure No.2

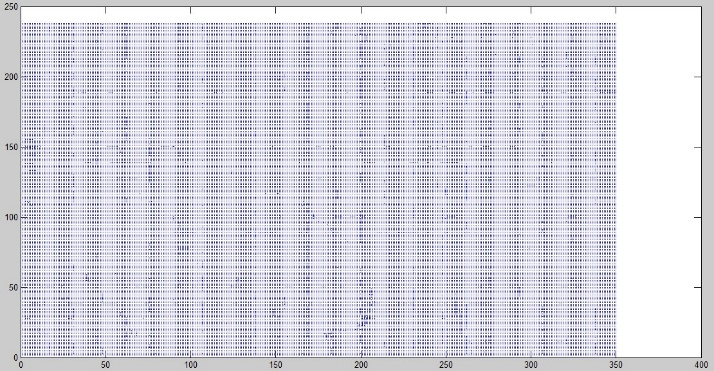
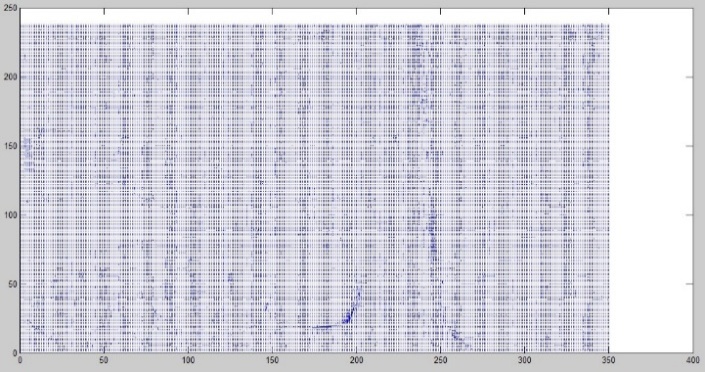
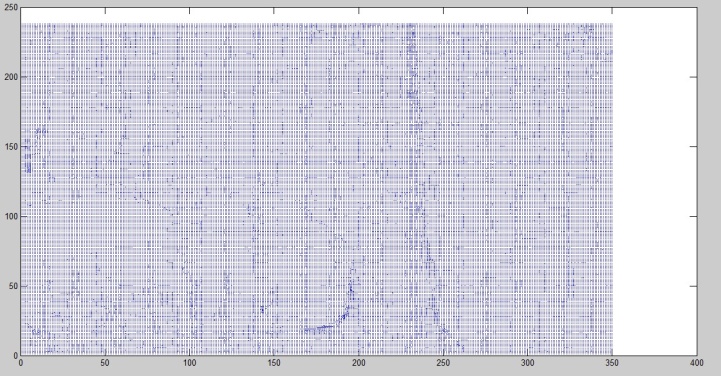
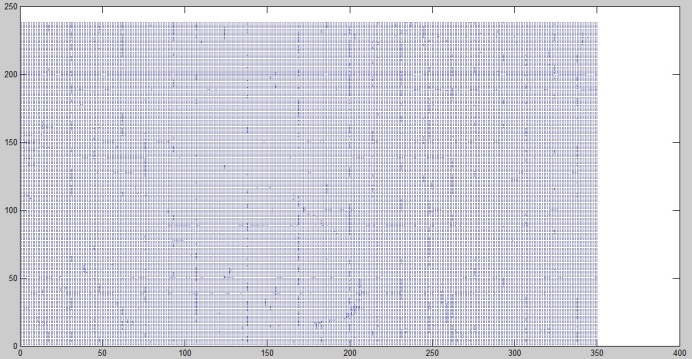


Figure No.3 Figure No.4



**FOR Video Servalance**

Figure No.1 Figure No.2

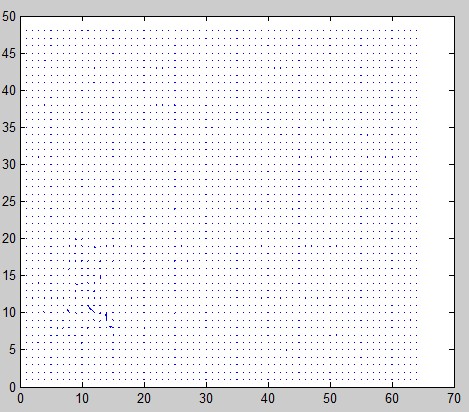
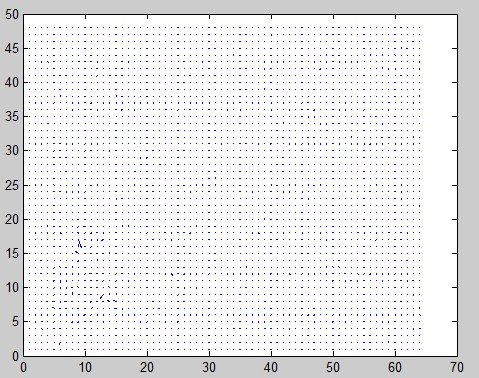


Figure No.3 Figure No.4

