Gunjan K Khut
UID – 115813083

Homework – Optical Flow Estimation.

The objective of the homework is to compute and compare optical flow on some sequences of the Middlebury Optical Flow dataset.

Part 1 -

Lucas - Kanade Implementation Output –

In this method, the optical flow in a local neighborhood of the pixel under consideration is essentially considered to be constant. The method of least squares criterion is used to solve the optical flow equations for all the pixels in the neighborhood.

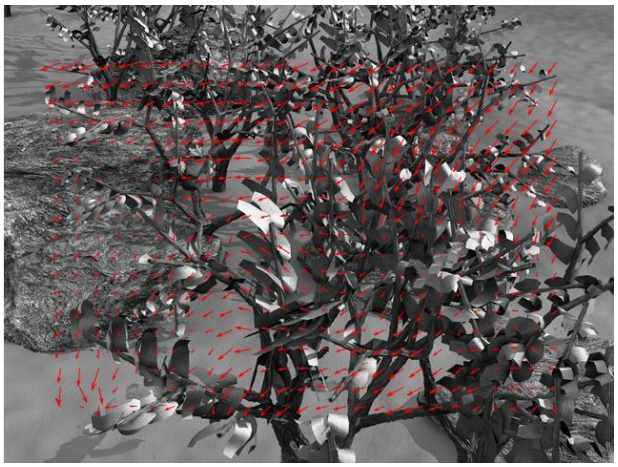

Fig 1. Output of Lucas Kanade Algorithm on Grove dataset

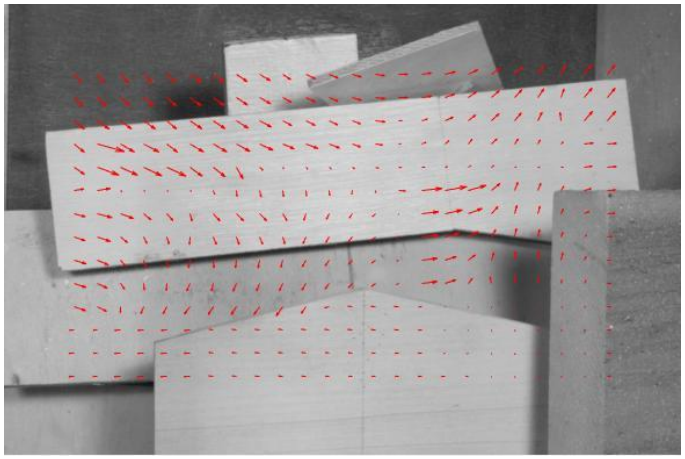

Fig 2. Output of Lucas Kanade Algorithm on Wooden dataset

Part 2 -

- **Horn and Schunck Implementation** – Horn and Schunck algorithm is based on a simple calculation. First, we need to compute the derivative along X and Y axis as well as time between the two images. Just by using convolution of 2x2 patches oriented in the axis we extract feature. We can extract the desired features:

$$Ix = ((Image1 * \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}) + (Image2 * \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}))/2$$

$$Iy = ((Image1 * \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}) + (Image2 * \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}))/2$$

$$It = ((Image1 * \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix}) + (Image2 * \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}))/2$$

From here we can imply our penalization to the X displacement matrix (u matrix) by λ and after this we just apply the equation corresponding to the algorithm following the number of iterations.

$$u = \bar{u} - Ix(It + \bar{v} * Iy + Ix * \bar{u})/(\lambda^2 + Iy^2 + Ix^2)$$

$$u = \bar{v} - Iy(It + \bar{u} * Iy + Ix * \bar{v})/(\lambda^2 + Iy^2 + Ix^2)$$

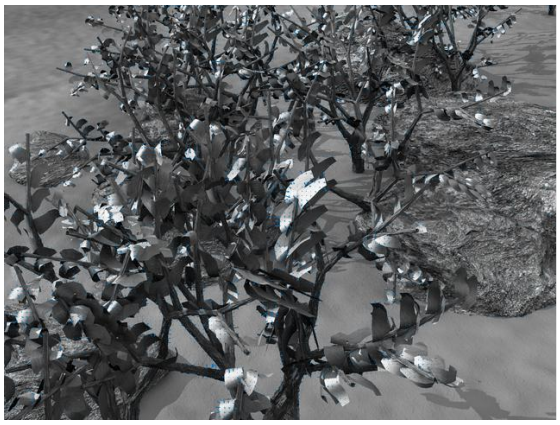

Fig 3. Output of MATLAB's function Horn-Schunck on Grove dataset

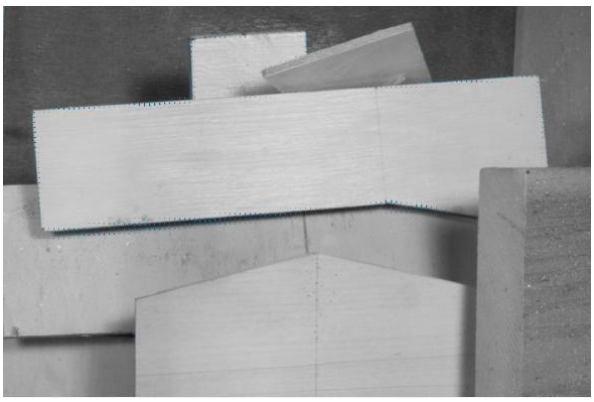

Fig 4. Output of MATLAB's function Horn-Schunck on Wooden dataset

- **<u>Lucas - Kanade Implementation</u>** – Lucas and Kanade process follow the same filtering process. The difference come from the algorithm equation and process, but we can already start directly after the filtering process. The main idea is to define a window size that will be used as a local patch to extract the optical flow and then combining these will gives us the result. These was the main lines. Taking care of this window size we can compute the u, v matrices just by simple computation such as resizing the vectors and computing a simple matrix calculus:

$$(A^{\mathrm{T}} * A)^{-1} * A^{\mathrm{T}} * B$$

With A corresponding to a concatenation of column transform of Ix and Iy, and B corresponding to column transformation of it.

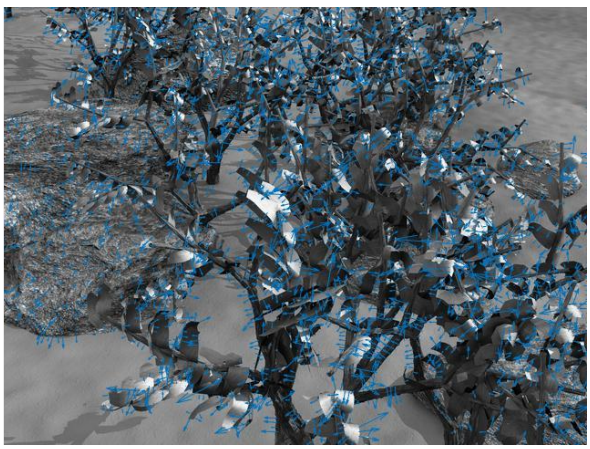

Fig 5. Output of MATLAB's function Lucas Kanade on Grove

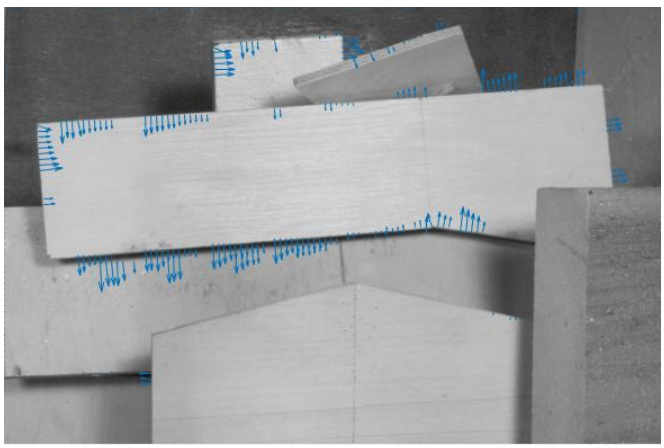

Fig 6. Output of MATLAB's function Lucas Kanade on Wooden dataset

Gunjan K Khut
UID – 115813083

- **Farneback Implementation** – Farneback method uses Polynomial Expansion to approximate the neighbors of a pixel. The Expansion is a quadratic equation with Matrices and Vectors as variable and coefficients. This dense optical flow analysis produces a displacement field from two successive video frames. Each displacement vector in the field is estimated by minimizing the 'error' under a 'constraint'.

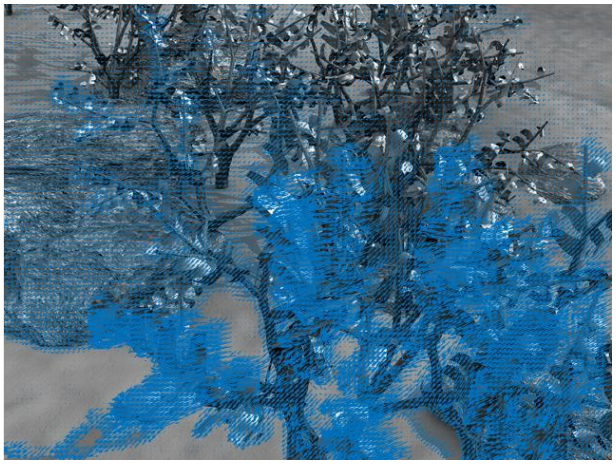

Fig 7. Output of MATLAB's function Farneback on Grove

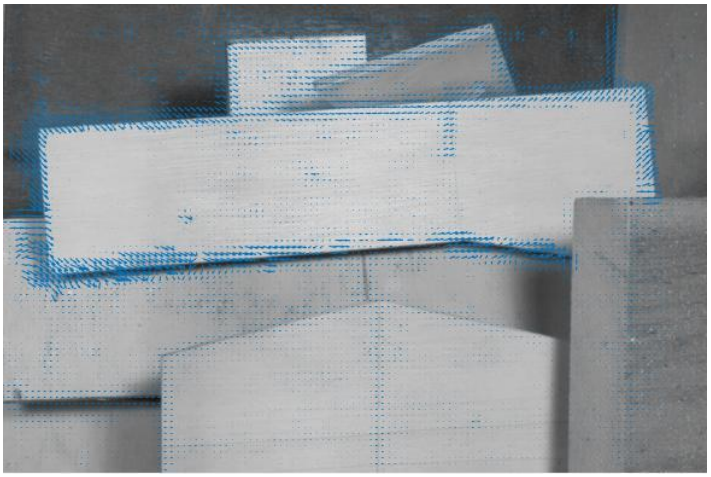

Fig 8. Output of MATLAB's function Farneback on Wooden dataset

- **Difference between the Algorithms by observations:**
    - The gradient-based algorithms typically performed better than the block-matching and variational methods. Horn-Schunck was able to produce lower absolute errors than Lucas-Kanade in most cases but proved to be more sensitive to parameter variations.
    - The wavelet noise pattern resulted in generally lower errors than either of the other backgrounds. In some cases where large errors were produced with the Gaussian background, the same algorithm and parameters were able to produce a good result when using the wavelet background instead.
    - All algorithms can produce severe errors with the wrong choice of parameter values. The choice is usually between excessive smoothing (large windows or high weights for regularization terms), or low robustness (small windows or low regularization weights).
    - Lucas Kanade is stable for smaller window sizes allowing for a better reproduction of high frequency detail.
    - The continuous parameters Horn-Schunck are more difficult to select
    - Horn & Schunk propagate smoothness across the entire image

- **Some other properties and differences:**

Horn-Schunck:

- The Horn–Schunck method of estimating optical flow is a global method which introduces a global constraint of smoothness to solve the aperture problem
- Horn Schunck method is effective in estimating the optical flow of boundary regions. It does not have consistent performance in estimating the optical flow of textured regions
- The Horn-Schunck algorithm delivered a low accuracy and a medium execution time.
- Advantages of the Horn–Schunck algorithm include that it yields a high density of flow vectors, i.e. the flow information missing in inner parts of homogeneous objects is filled in from the motion boundaries. On the negative side, it is more sensitive to noise than local methods.

Lucas Kanade -

- It is stable on corners because –
  - Gradients are different, large magnitudes
  - Large λ1, large λ2
- Unstable on low texture patches because –
  - Gradients have small magnitude
  - Small λ1, small λ2
- Unstable on edges –
  - large gradients in one direction
  - large λ1, small λ2
- Run time of lucas Kanade is the least of all.
- This method is effective in estimating the optical flow of textured regions, and good in estimating the optical flow of textured regions.
- The Lucas Kanade is sparse flow estimator, and do not estimate flow for
- It is the least accurate of all
- The nearest flow result delivers the best endpoint and angular errors, with a medium execution time, but the density is very low. There are just a few points included in the measurement which are normally located in the same area of the image and not distributed over the whole image
- The potential causes of errors of Lucas Kanade are
  - ATA is not easily invertible
  - Noise in the image
  - When our assumptions are violated
  - Brightness constancy is not satisfied
  - The motion is not small
  - A point does not move like its neighbors (as window size is too large)

Gunjan K Khut
UID – 115813083

Farneback:

- Farneback has the highest runtime
- Farneback method appears to be more sensitive in object boundaries than the textured regions.
- Farneback algorithm has the best trade-off between accuracy and execution time

Conclusion:

- The short explanation is, sparse techniques only need to process some pixels from the whole image, dense techniques process all the pixels. Dense techniques are slower but can be more accurate.

References:

- https://en.wikipedia.org/wiki/Horn–Schunck_method
- https://en.wikipedia.org/wiki/Lucas–Kanade_method
- MathWorks MATLAB Functions
- https://www.hsr.ch/uploads/tx_icscrm/17_DAB_E_high_13.pdf
- https://scholarsarchive.byu.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=2670&context=etd
- http://www.cs.umd.edu/~djacobs/CMSC426/OpticalFlow.pdf
- http://ijettjournal.org/volume-4/issue-10/IJETT-V4I10P142.pdf
- Class lectures and presentations on optical flow estimation
- UCF Computer Vision Video Lectures 2014 - YouTube