Eighth report - 2D Motion Estimation

Kimberly Grace Sevillano C.

***Subject*— Implementing 2D Motion Estimation technique.**

I. Objective

The goal of this report is to select and test a method: image difference, optical flow, or block correlation, on images with either camera or object movement.

A. Problem Description

Motion estimation tracks object movements in image sequences for applications like video compression and object tracking. Traditional methods may be inaccurate, while optical flow estimates pixel-wise motion based on intensity changes, assuming constant object brightness.

II. Methods

In this section, we examine three distinct methods for 2D motion estimation: image difference, optical flow, and block correlation. Each technique has its unique advantages and disadvantages in various applications.

A. Image Difference

Image difference is a basic motion estimation method, calculating absolute differences between consecutive frames. Its simplicity ensures fast computation but struggles with complex motion, noise, and illumination changes.

B. Block Correlation

Block correlation, a more advanced motion estimation method, compares pixel blocks between frames to find the best match and calculate motion vectors. It's more robust to noise and illumination changes but can be computationally expensive for large search regions and high-resolution videos.

C. Optical Flow

Optical flow estimates motion in image sequences by analyzing brightness patterns, assuming pixel intensity remains constant between frames. It captures complex motion patterns but can be sensitive to noise and less accurate in low-texture or non-rigid motion regions.

III. Methodology

The methodology for implementing this report is Optical Flow:

A. Optical Flow

Optical flow is an advanced motion estimation method that estimates the motion of objects or the camera based on the intensity patterns in a video sequence. In this study, we implement the Lucas-Kanade method, which assumes that the flow is essentially constant within a small window of pixels.

The optical flow method is implemented in the optical_flow function. This function takes two grayscale images and a window size as input. It calculates the gradients in the x, y, and t (time) directions using convolution with different kernels. It then iterates through the pixels of the images, taking a small window around each pixel. For each window, the motion vector components are calculated if the determinant of a specific matrix A is greater than a small threshold value. The function returns the horizontal and vertical components of the motion vectors for each pixel.

The Lucas-Kanade method can be represented by the following equation:

$$A \begin{bmatrix} u & v \end{bmatrix} = -b \tag{1}$$

Where:

- A is a matrix of size 2×2, formed by the sum of the products of the gradients in the x and y directions within the window.
- u and v are the horizontal and vertical components of the motion vector, respectively.
- b is a vector of size 2, formed by the sum of the products of the gradients in the x and y directions, and the gradient in the time direction within the window.

The components of A and b can be calculated as follows:

$$A = \sum_W \begin{bmatrix} I_x^2 & I_x I_y \; I_x I_y & I_y^2 \end{bmatrix} \tag{2}$$

$$b = \sum_W \begin{bmatrix} I_x I_t \; I_y I_t \end{bmatrix} \tag{3}$$

Where:

- I_x, I_y, and I_t are the gradients in the x, y, and time directions, respectively.
- W is the window around the pixel.

The provided pseudocode details the steps in the optical_flow function:

Algorithm 1 Lucas-Kanade Optical Flow

```
Require: Grayscale images $img1$, $img2$, window size $window\_size$
Ensure: Optical flow vectors $[u, v]$
 1: Initialize: Set $\tau \leftarrow 10^{-2}$, set kernels $x_{kernel}$, $y_{kernel}$, and $t_{kernel}$
 2: Preprocessing: Normalize $img1$ and $img2$ by dividing by 255, compute gradients $gradx$, $grady$, and $gradt$ using 2D convolution with kernels
 3: Compute Flow: Initialize $u$ and $v$ as zero matrices with the same shape as $img1$
 4: for each pixel $(i, j)$ in the images, considering the window size do
 5:     Compute $Ix$, $Iy$, and $It$ using gradient matrices
 6:     Compute $sumxx$, $sumxy$, $sumyy$, $sumxt$, and $sumyt$
 7:     Form matrix $A$ and vector $b$
 8:     Compute SVD of $A^T A$: $U$, $D$, $V^T$
 9:     if $\min(D) \geq \tau$ then
10:         Compute $nu \leftarrow A^{-1}b$
11:         Update optical flow vectors: $u[i, j] \leftarrow nu[0]$, $v[i, j] \leftarrow nu[1]$
12:     else
13:         Set optical flow vectors to 0: $u[i, j] \leftarrow 0$, $v[i, j] \leftarrow 0$
14:     end if
15: end for
16: return $[u, v]$
```

IV. Results

The Optical Flow method was applied to various image pairs to analyze its performance in estimating motion. For each image pair, we compared the original image to the optical flow visualization, consisting of arrows representing the estimated motion vectors. The results show different motion patterns, including object movement, camera movement, and a combination of both.

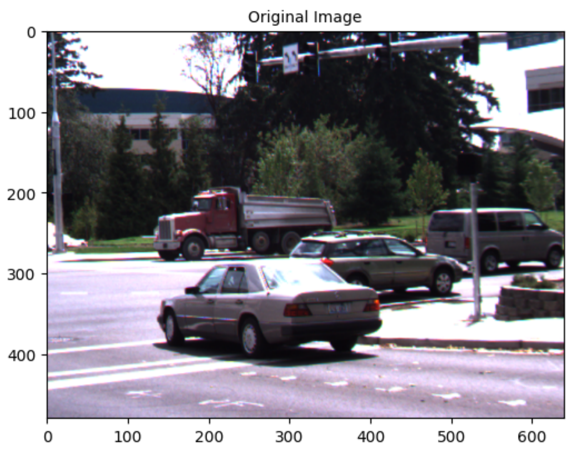

Fig. 1: Original image for Example 1.

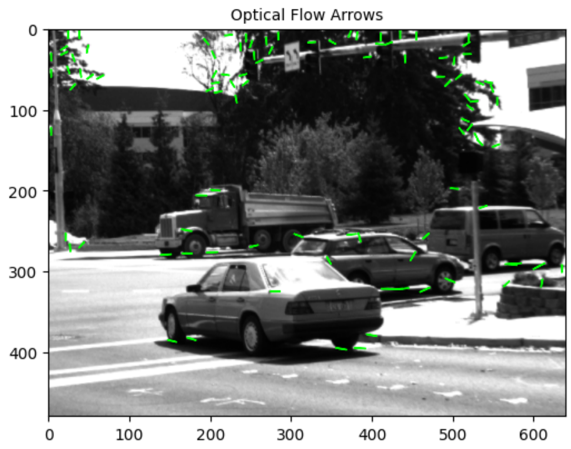

Fig. 2: Optical Flow arrows for Example 1.

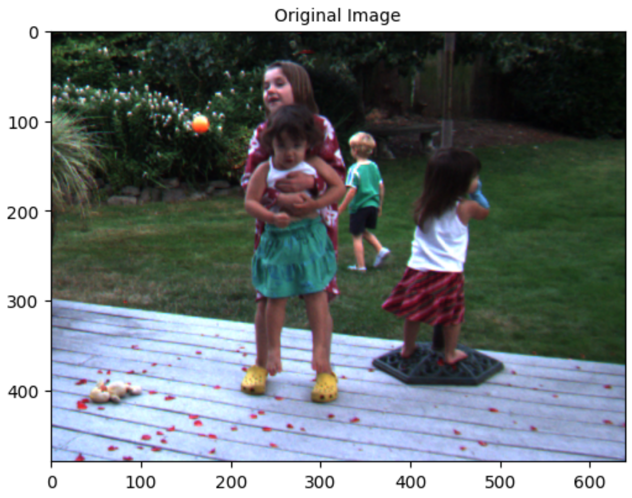

Fig. 3: Original image for Example 2.

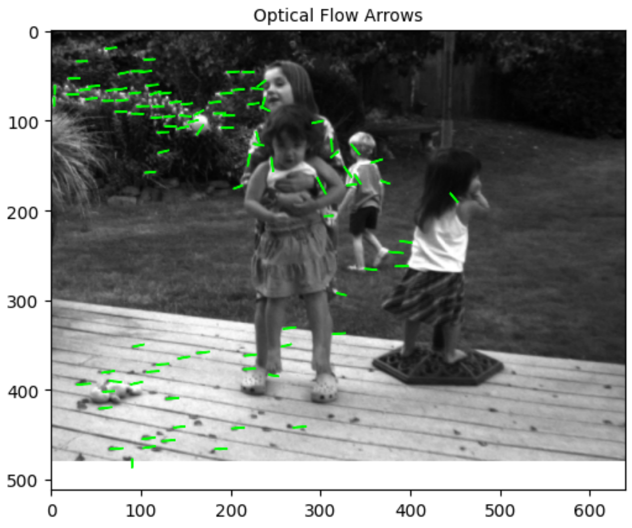

Fig. 4: Optical Flow arrows for Example 2.

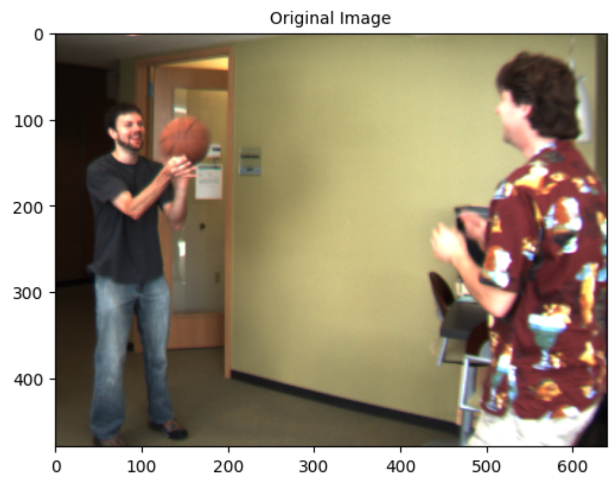

Fig. 5: Original image for Example 3.

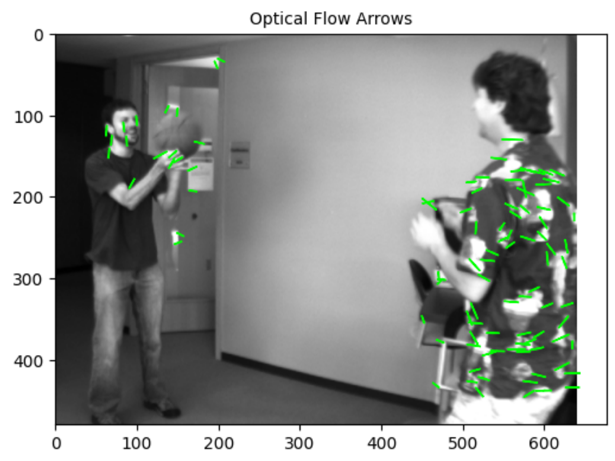

Fig. 6: Optical Flow arrows for Example 3.

In the Examples 1, 2, 3 the Optical Flow method detected object movement in the scene. The arrows in the optical flow visualization (Figures 2, 4, 6) indicate the direction and magnitude of the estimated motion.

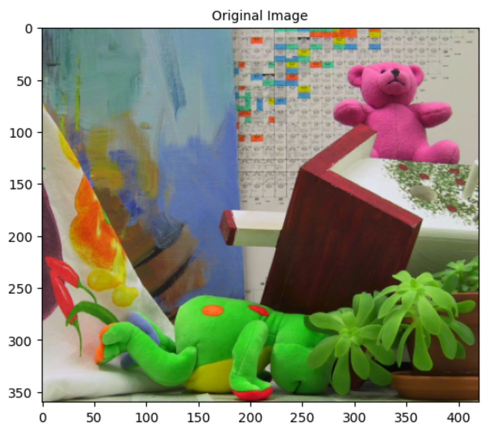

Fig. 7: Original image for Example 4.

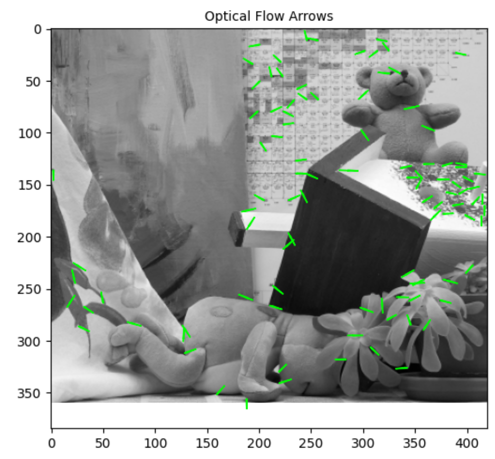

Fig. 8: Optical Flow arrows for Example 4.

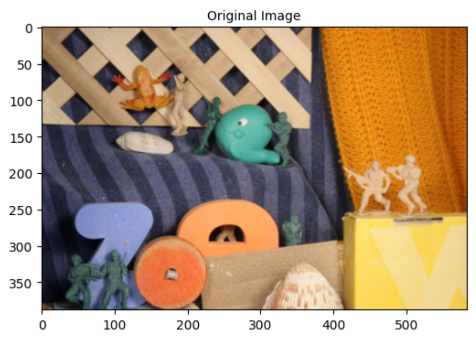

Fig. 9: Original image for Example 5.

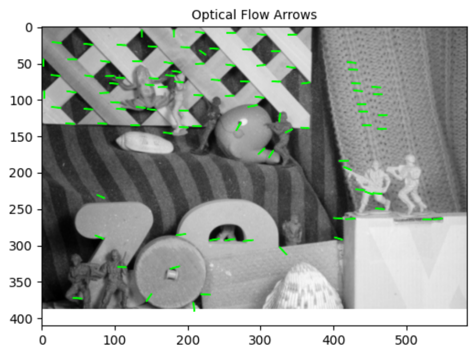

Fig. 10: Optical Flow arrows for Example 5.

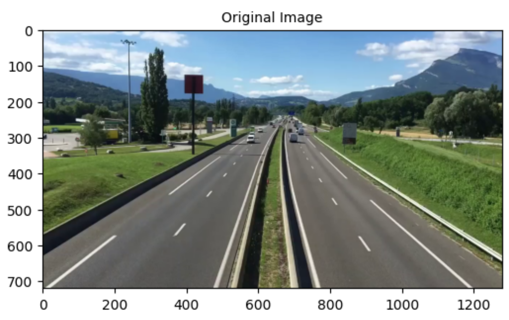

Fig. 11: Original image for Example 6.

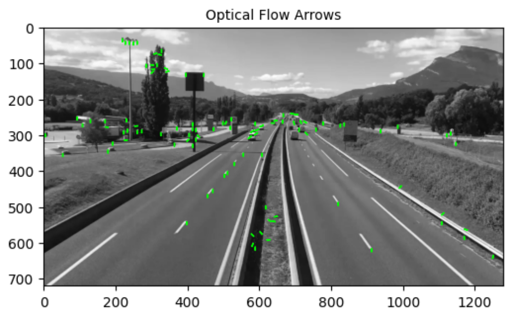

Fig. 12: Optical Flow arrows for Example 6.

In the Examples 4, 5, 6 the Optical Flow method detected camera movement. The arrows in the optical flow visualization (Figures 8, 10, 12) show the overall motion of the scene, indicating that the camera was moving during image capture.

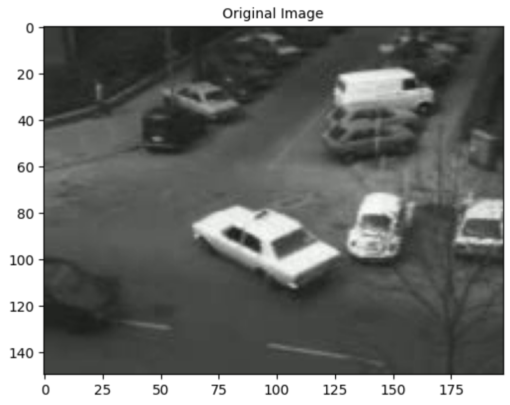

Fig. 13: Original image for Example 7.

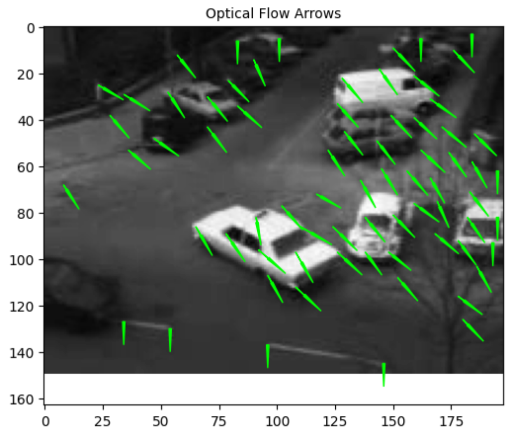

Fig. 14: Optical Flow arrows for Example 7.

In the Example 7, the Optical Flow method detected a combination of object and camera movement. The arrows in the optical flow visualization (Figure 14) represent a complex motion pattern, indicating both object and camera movements within the scene.

These results demonstrate the effectiveness of the Optical Flow method in estimating different types of motion in image sequences. The method successfully detected object movement, camera movement, and a combination of both, providing valuable insights for applications such as video compression and object tracking.

V. CONCLUSIONS

In this report, we have implemented and analyzed the Optical Flow method for 2D motion estimation using the Lucas-Kanade algorithm. The method was applied to various image pairs, and the results demonstrated its effectiveness in estimating different types of motion, including object movement, camera movement, and a combination of both.

The Optical Flow method provides valuable insights for applications such as video compression and object tracking. Its ability to capture complex motion patterns makes it a powerful tool for analyzing motion in image sequences. However, it can be sensitive to noise and less accurate in low-texture or non-rigid motion regions. Future work may include exploring other Optical Flow algorithms and addressing these limitations to improve motion estimation accuracy in different scenarios.

REFERENCES

[1] Lucas, B. D., & Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, 674–679

[2] Horn, B. K. P., & Schunck, B. G. (1981). Determining Optical Flow. Artificial Intelligence, 17(1-3), 185–203.

[3] Bouguet, J.-Y. (2000). Pyramidal Implementation of the Lucas Kanade Feature Tracker: Description of the algorithm. Intel Corporation, Microprocessor Research Labs.

[4] Barron, J. L., Fleet, D. J., & Beauchemin, S. S. (1994). Performance of Optical Flow Techniques. International Journal of Computer Vision, 12(1), 43–77.