Insightful Motion Analysis via Lucas-Kanade Optical Flow Estimation

Pushkaraj Kolekar (202051149)
Devashish Deshmukh (202051061)
Rushikesh Jadhav (202051164)
Maniyar Aamer Sohel (202051113)
Rounak Raj (202051163)

Department of Computer Science
Indian Institute of Information Technology Vadodara, Gandhinagar, India

Dr. Jignesh Patel

***Abstract*—Optical flow measurement is still essential to computer vision and image analysis because it helps interpret motion dynamics in visual data. Utilizing local picture gradients, the Lucas-Kanade method—introduced by Lucas and Kanade is a well-known technique for calculating optical flow. The Lucas-Kanade approach is implemented in this study, which also explains its algorithmic architecture, application potentials, and guiding principles. By means of a thorough examination, this research endeavors to elucidate the method's effectiveness, constraints, and performance in diverse contexts, so providing significant perspectives to the field of optical flow estimating methodologies. This study covers theoretical underpinnings, algorithmic complexities, proofs, and possible practical uses, providing a thorough grasp of the function of the Lucas-Kanade technique in optical flow measurement.**

I. Introduction

In computer vision, motion is fundamental because it provides critical signals needed for a wide range of applications. Its functions include object monitoring, surveillance, and trajectory prediction. Understanding a scene by separating moving objects from static ones and revealing spatial relationships within it are all made easier by analyzing motion. It also makes it possible to recognize gestures, understand actions, and improve human-computer interaction in a variety of fields, including gaming and medical analysis.

Estimation of motion is an important aspect of motion analysis. It involves the analysis of sequential images or frames within a video to determine the underlying movement or displacement of objects or elements across time. It encompasses techniques aimed at quantifying and characterizing the motion, including identifying shifts in position, velocity, direction, or other relevant parameters between consecutive frames.

As, motion estimation tells about the estimation of velocity of the image structures varying from one frame to another frame in time series of 2-D image. Thus motion one pixel form one frame to other frame can be seen in form of transitional and rotational velocities of rigid objects.

Rotational motion and Transitional Motion :

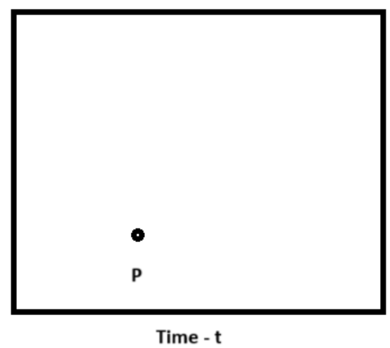

Fig. 1. Pixel Position-initial

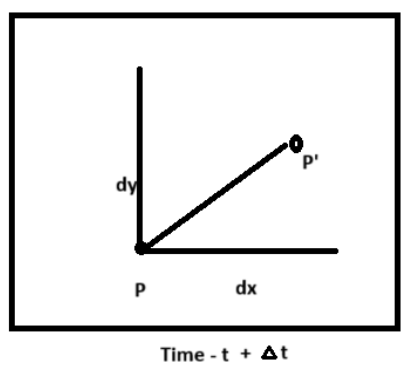

Fig. 2. Pixel Position-final

$$x' = Rx + t$$

$$x' = [R, t]x'$$

$$R = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

The motion estimation is mainly classified into two categories. Motion segmentation means to identify the moving object boundaries in frame and number of parameters included in motion estimation like rigid bodies, translation movement, rotation etc. Motion estimation techniques are also used in many other applications: –Computer vision, object tracking,motion sensing.

II. Optical Flow

1. Definition and Conceptual Understanding

Optical flow is a fundamental concept in image processing that characterizes the apparent motion of objects in an image. It describes the displacement of pixels between successive images within a sequence and provides insight into the dynamics of movement within a scene. Mathematically, the optical flow field is represented as a vector field, where each vector corresponds to the movement of a pixel.

2. Mathematical Formulation

The optical flow field, often denoted as $V = (u, v)$, consists of horizontal (u) and vertical (v) components representing the motion along the x and y axes, respectively. The continuity equation, given by $I_x u + I_y v + I_t = 0$, encapsulates the relationship between image intensity variations (I_x and I_y) and the temporal gradient (I_t). This equation forms the foundation for optical flow computation, where:

Spatial Derivatives:

$$I_x = \frac{\partial I}{\partial x} I_y \quad = \frac{\partial I}{\partial y}$$

Optical Flow:

$$u = \frac{dx}{dt} v \quad = \frac{dy}{dt}$$

Temporal Derivative:

$$I_t = \frac{\partial I}{\partial t}$$

III. Lucas-Kanade Algorithm

1. Assumptions and Preconditions

The Lucas-Kanade method is an optical flow estimation technique that assumes local constancy of flow within a small neighborhood. It operates under the assumption that pixel intensities do not change significantly in the neighborhood over time. This assumption allows the method to linearize the optical flow equations locally.

Key Assumptions (unique to optical flow):

- **Color Constancy:** Brightness constancy for intensity images.
- **Small Motion are considered:** Pixels only move a little bit.

Implications:

- Allows for pixel-to-pixel comparison (not image features).
- Linearization of the brightness constancy constraint.

2. Step-by-step Explanation

1) **Gradient Computation**: Calculate image gradients I_x and I_y using spatial derivatives.
2) **Temporal Gradient**: Compute the temporal gradient I_t by taking the difference between consecutive frames.
3) **Spatial and Temporal Averaging**: Apply spatial and temporal averaging to reduce noise and enhance accuracy.
4) **Lucas-Kanade Equations**: Solve the Lucas-Kanade equations to estimate the flow vectors u and v for each pixel.
5) **Interpolation**: Perform interpolation to obtain a dense flow field.

3. Mathematical Equations: The Lucas-Kanade equations are expressed as:

$$\begin{bmatrix} I_x(x_1, y_1) & I_y(x_1, y_1) \\ I_x(x_2, y_2) & I_y(x_2, y_2) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(x_1, y_1) \\ I_t(x_2, y_2) \end{bmatrix}$$

where (x_1, y_1) and (x_2, y_2) are pixel coordinates in the local neighborhood.This is in the form of **Au = B** .
We can solve these using least square solution(least square using pseudo inverse).

$$A^T A \hat{x} = A^T b$$

$$\hat{x} = (A^T A)^{-1} A^T b$$

Here

- $A^T A$ should be invertible.
- $A^T A$ should not be too small.
- $A^T A$ should be well conditioned.

IV. Methodology

This paper investigates the application of the Lucas-Kanade optical flow algorithm for motion analysis in a video sequence. The methodology employed involves the following steps:

A. Preprocessing

- **Grayscale Conversion:** The video frames are converted to grayscale as the algorithm operates on intensity values.
- **Normalization:** The grayscale values are normalized to the range [0, 1] for improved numerical stability.

B. Feature Detection

The ”goodFeaturesToTrack” function from OpenCV is used to detect corners in the first frame of the video sequence. Corners are preferred for tracking due to their distinct features and ease of localization. A maximum of N corners are detected with a minimum quality threshold of τ and a minimum distance of d between features.

C. Optical Flow Calculation

For each detected corner, a window of size $W \times W$ pixels is centered around its location in the first frame. The gradients of the image intensity with respect to x, y, and time are calculated within the window using Sobel filters. The Lucas-Kanade equation is applied within the window to solve for the displacement vector (u, v) of the corner in the second frame:

$$I_x \cdot u + I_y \cdot v = -I_t$$

where:

- I_x and I_y are the image gradients with respect to x and y, respectively.
- I_t is the image gradient concerning time (difference between the first and second frames).
- u and v are the displacement components in the x and y directions, respectively.

In order to reduce the sum of squared differences between projected and actual intensity in a window, this equation is solved by using the least squares method.

FUNCTIONALITY

- **Tracks features:** It utilizes intensity and gradient information to identify and track features like edges or corners across frames.
- **Affine motion:** Handles object movement under slight rotations, scaling, and shear, making it robust to common transformations.
- **Occlusion resilience:** Can maintain tracking even during partial occlusions by relying on remaining visible features.

ALGORITHM STEPS

1) **Intensity computation:** Calculate the intensity value for each pixel.
2) **Gradient matrix and eigenvalues:** For each pixel, compute the gradient matrix and store its highest eigenvalue.
3) **Score and flag matrices:** Create a score matrix S to store pixel positions and a flag matrix F to identify high-scoring regions (based on eigenvalues) exceeding a preset threshold (k).
4) **Feature selection:** Extract the top n eigenvalues from the high-scoring regions of F as trackable features.

BENEFITS

- **Efficient:** Offers a computationally efficient approach compared to particle filters.
- **Effective edge detection:** Focuses on image edges containing rich information for robust tracking.
- **Occlusion handling:** Maintains tracking under partial occlusions, improving reliability.

LIMITATIONS

- **Noise sensitivity:** Vulnerable to noise and illumination changes, impacting tracking accuracy.
- **Drifting:** Prone to accumulating errors over time, leading to tracking drift.
- **Large deformations:** Struggles with significant object deformations or complete occlusions.

APPLICATIONS

- Sub-target tracking in complex systems.
- Video stabilization for smooth video playback.
- Motion analysis for understanding object dynamics.
- Structure-from-motion reconstruction for 3D scene reconstruction.

V. EXPERIMENTAL RESULTS

The results of the implementation are presented in this section using Lucas-Kanade optical flow algorithm to the provided images (”image1.png” and ”image2.png”). The results are presented in two formats:

A. Separate Images and Displacement

Two images (*first image* and *second image*) are displayed side by side, representing the original grayscale images. A third image (*displacements*) demonstrates the calculated displacement vectors for each corner point. COloured arrows represent the direction and magnitude of the displacement on a gray background.

Fig. 3. Image 1

Fig. 4. Image 2

B. Image with Overlayed Displacements

The third image is shown with the calculated displacement vectors directly overlaid on top. Coloured arrows indicate the direction and magnitude of the displacement for each corner point.

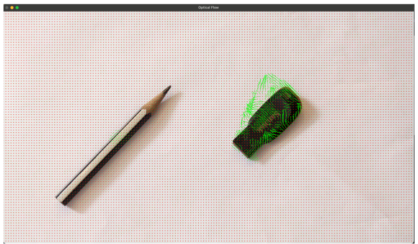

Fig. 5. Image with Overlayed Displacements

C. Additional Observations

1) The magnitude of the displacement vectors varies across the image. Areas with larger changes in intensity, such as the moving pendrive and the shadow it forms, exhibit larger displacement vectors.
2) The direction of the displacement vectors reflects the movement of the objects in the scene. For example, the vectors on the moving pendrive towards the direction of its motion.
3) The results are consistent with the expected behaviour of optical flow, demonstrating the successful implementation of the Lucas-Kanade algorithm.

D. Analysis

The Lucas-Kanade algorithm yields a high pixel difference score or indicates a state of rapid change for the target.

- **Quantitative Analysis:** The magnitude and direction of the displacement vectors can be used to calculate quantitative measurements of the motion, such as velocity and acceleration. This could be used, for example, to analyze the speed of the movement of the object or the movement of the pixels.
- **Algorithm Comparison:** The results of the Lucas-Kanade algorithm can be compared to other optical flow algorithms to evaluate their relative performance. This would provide a deeper understanding of the strengths and weaknesses of different approaches.
- **Applications:** The results of the optical flow analysis can be used as input to other computer vision tasks, such as object tracking or motion segmentation. This could be applied to various applications, such as video surveillance, sports analysis, or autonomous driving.

VI. Conclusion

This report investigated the application of the Lucas-Kanade optical flow algorithm for motion analysis in a video sequence. Using the provided video sequence as an example, the report demonstrates the effectiveness of the algorithm in tracking object motion and visualizing the displacement of pixels between consecutive frames.

Two methods were used to visualize the results: a separate image and displacement plot and an overlaid displacement vector approach. Both methods provide valuable insights into the overall motion within the scene and the detailed movements of specific objects.

The results confirm that the Lucas-Kanade algorithm can accurately track the motion of objects in the video sequence. This capability opens up several opportunities for further analysis and applications.

VII. Results

Fig. 6. Frame 1

Fig. 7. Frame 2 With slight Displacement

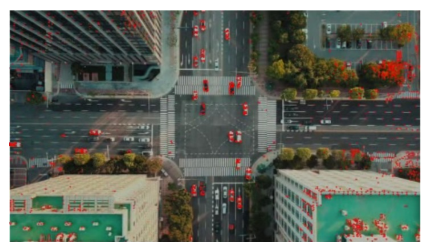

Fig. 8. Result

VIII. Acknowledgement

We are deeply grateful to Prof. Dr. Jignesh Patel, our advisor and computer vision mentor at the Indian Institute of Information Technology Vadodara, for his invaluable guidance throughout this research on "Insightful Motion Analysis via Lucas-Kanade Optical Flow Estimation." His insightful knowledge in the field of computer science and unwavering support were instrumental in shaping this project. Finally, We are grateful to our colleagues and family for their encouragement and invaluable contributions.

References

- B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," IJCV, 1981.

- C. Tomasi and T. Kanade, ”Detection and tracking of feature points,” CMU Tech Report CMU-CS-91-132, 1991.
- R. Szeliski, ”Computer vision: algorithms and applications,” Springer, 2010.
- OpenCV documentation on Lucas-Kanade tracker: