University Of Maryland

College Park

Project Report

673-Perception for Autonomous Robots

Project 4

Aditya Goswami
116951968

Mahmoud Dahmani
116777896

Sukoon Sarin
116955522

April 6, 2020

Contents

List of Figures

1 Introduction

In this project we had to implement the Lucas-Kanade template tracker. Then evaluated our code on three video sequences from the Visual Tracker benchmark database: featuring a car on the road, a baby fighting a dragon, and Usain Bolt's race.

Optical flow of events inside a frame is found by Lucas Kanade Algorithm. These events in a frame are gathered after considering a pixel and its adjacent pixels. The least square criterion is used to approximate the solution. The change of an object in the image and its pixel are gone through for two frames and at any moment of time, they are nearly the same. For optical flow, this same pixel intensity between two frames is used to find the flow of motion or the vector flow. Following this, the optical flow equations are used in the pixels and the window center.

- Lucas Kanade is a method to generate the optical flow of events in a frame.
- These events are gathered after considering a pixel and its neighbouring pixels.
- The least square criterion, an approach in regression analysis is used to approximate the solution. The displacement of an image and its pixel contents are studied for two frames and at these two instants these are nearly the same.
- Thus, optical flow equations are used to and assumed to be true for all the pixels in a window center.
- Further below is the detailed report of our steps for each problem.

2 Data Preparation

1. There was no need to prepare any specific data set for this project.
2. We were provided three data set already by the professor and the TAs.
3. The three video sequences (or data set of images of the images) were already prepared and given in an arranged order.
4. Namely the three data set provided and used in this project are -
5. a car on the road
6. a baby fighting a dragon
7. Usain Bolt's race

(a) Car on Road

(b) Baby Fighting Dragon

(c) Usain Bolt's Race

Figure 1: Three Data sequences

3 Lucas Kanade Algorithm

In computer vision, the Lucas–Kanade method is a widely used differential method for optical flow estimation developed by Bruce D. Lucas and Takeo Kanade.

Optical flow or optic flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene.

Optical flow can also be defined as the distribution of apparent velocities of movement of brightness pattern in an image.

The concept of optical flow was introduced by the American psychologist James J. Gibson in the 1940s to describe the visual stimulus provided to animals moving through the world.

Gibson stressed the importance of optic flow for affordance perception, the ability to discern possibilities for action within the environment. Followers of Gibson and his ecological approach to psychology have further demonstrated the role of the optical flow stimulus for the perception of movement by the observer in the world; perception of the shape, distance and movement of objects in the world; and the control of locomotion.

The term optical flow is also used by roboticists, encompassing related techniques from image processing and control of navigation including motion detection, object segmentation, time-to-contact information, focus of expansion calculations, luminance, motion compensated encoding, and stereo disparity measurement.

It assumes that the flow is essentially constant in a local neighbourhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighbourhood, by the least squares criterion.

By combining information from several nearby pixels, the Lucas–Kanade method can often resolve the inherent ambiguity of the optical flow equation. It is also less sensitive to image noise than point-wise methods. On the other hand, since it is a purely local method, it cannot provide flow information in the interior of uniform regions of the image.

4 Least Square Criterion and Affine Parameters

4.1 Least Square Criterion

'u' and *'v'* are the hypothesized location of template in current frame which is given by:

$$E(u,v) = \sum [I(x+u, y+v) - T(x,y)]^2$$

$$E(u,v) \approx \sum [I(x,y) + uI_x(x,y) + vI_y(x,y) - T(x,y)]^2$$

$$E(u,v) = \sum [uI_x(x,y) + vI_y(x,y) + D(x,y)]^2$$

By taking partial derivatives and equating to zero:

$$\partial E/du = \sum [uI_x(x,y) + vI_y(x,y) + D(x,y)]I_x(x,y) = 0$$

$$\partial E/dv = \sum [uI_x(x,y) + vI_y(x,y) + D(x,y)]I_y(x,y) = 0$$

Form matrix equation which is solved using Least-Squares method:

$$\sum \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\sum \begin{bmatrix} I_xD \\ I_yD \end{bmatrix}$$

4.2 Least Square: Affine Parameters

The Warp $W(x;p)$ takes the values of the pixels x in the coordinate frame and for the template T it maps to the every pixel's area $W(x;p)$ in the coordinate frame I. $W(x;p)$ denotes the set of parameters $p = (p_1, ..., p_n)^T$ is the vector of parameters.

$$W(x;p) = \begin{pmatrix} x + p_1 \\ y + p_2 \end{pmatrix}$$

The optical flow is now the vector of parameters $p = (p_1, p_2)^T$. To calculate Affine warps:

$$W(x;p) = \begin{pmatrix} (1+p_1) \cdot x + p_3 \cdot y + p_5 \\ p_2 \cdot x + (1+p_4) \cdot y + p_6 \end{pmatrix} = \begin{pmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

5 Lucas Kanade Algorithm and the steps involved

5.1 Lucas Kanade Algorithm

In this expression, $\bigtriangledown I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)$ is the gradient of image I evaluated at $W(x;p)$; i.e. $\bigtriangledown I$ is computed in the coordinate frame of I and then warped back onto the coordinate frame of T using the current estimate of the warp $W(x;p)$. The term $\frac{\partial W}{\partial p}$ is the Jacobian of the warp. If the warp which is $W(x;p) = (W_x(x;p), W_y(x;p))^T$ then:

$$\frac{\partial W}{\partial p} = \begin{pmatrix} \frac{\partial W_x}{\partial p_1} & \frac{\partial W_x}{\partial p_2} & \cdots & \frac{\partial W_x}{\partial p_6} \\ \frac{\partial W_y}{\partial p_1} & \frac{\partial W_y}{\partial p_2} & \cdots & \frac{\partial W_y}{\partial p_6} \end{pmatrix}$$

We follow the notational convention that the partial derivatives with respect to a column vector are laid out as a row vector. This convention has the advantage that the chain rule results in a matrix multiplication. For example, the affine warp has the Jacobian:

$$\frac{\partial W}{\partial p} = \begin{pmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{pmatrix}$$

By performing the first order Taylor expansion

$$\sum_x \left[I\left(W\left(x;p\right)\right) + \bigtriangledown I \frac{\partial W}{\partial p} \Delta p - T\left(x\right) \right]^2$$

Setting this expression to equal zero after multiplying hessian matrix and solving gives the closed form solution as:

$$\Delta p = H^{-1} \sum_x \left[\bigtriangledown I \frac{\partial W}{\partial p} \right]^T \left[T(x) - I(W(x;p)) \right]$$

The Hessian marix is calculated as followed

$$H = \sum_x \left[\bigtriangledown I \frac{\partial W}{\partial p} \right]^I \left[\bigtriangledown I \frac{\partial W}{\partial p} \right]$$

5.2 The Lucas Kanade Steps

Iterate

(a) Warp I with $W(x;p)$ to compute $I(W(x;p))$

(b) Compute the error in the image $T(x) - I(W(x;p))$

(c) Warp the gradient $\bigtriangledown I$ of the template $W(x;p)$

(d) Evaluate the Jacobian $\frac{\partial W}{\partial p}$ at $(x;p)$

(e) Compute the steepest descent images $\bigtriangledown I \frac{\partial W}{\partial p}$

(f) Compute the Hessian Matrix.

(g) Compute $\sum_x \left[\bigtriangledown T \frac{\partial W}{\partial p}\right]^T [T(x)) - I(W(x;p))]$

(h) Compute Δp

(i) Update the parameters $p \leftarrow p + \Delta p$

untill $\|\Delta p\| \leq \epsilon$

6 Pipeline

(a) The algorithm is initialized by defining a template around the object to be tracked.

(b) Parameters are updated in the current frame ($p + \Delta p$)

- For the 'baby fighting the dragon' this works better.
- For the car, due to the sudden change in intensity, Histogram equalization is used. In addition to this, features are extracted at every 100^{th} frame again that are to be tracked in subsequent frames.
- For race, again due to less difference between intensities it tracks closest another athlete.

(c) These parameters help draw the bounding box.

7 Analysis

(a) 0.001, 0.001 and 0.0001 is used for Δp in human, car and vase as threshold.

(b) Robust results are calculated from Lucas-Kanade Inverse Compositional Algorithm variation is used as on implementation.

(c) For robustness, the features are fed in again after a couple of frames.

7.1 Tracker Evaluation

(a) During our implementation we found that for car, Baby fighting, and Bolt's race we need tight bounding boxes because these objects are moving, and the changes in background creates noise. Hence, we want the least portion of the background.

(b) The tracker fails when the object is moving at high speeds. In case of fast-moving objects, the optical flow vector becomes too large in subsequent frames and hence the LK tracker can't track it. The tracker also fails or underperform in case there's a change in illumination. Here the algorithm fails because the sum of squared distances error that it tries to minimize is sensitive to illumination changes.

7.2 Robustness

(a) The LK tracker as it is formulated, breaks down when there is a change in illumination because the sum of squared distances error that it tries to minimize is sensitive to illumination changes. There are a few things we did to fix this.

(b) First we scaled the brightness of pixels in each frame so that the average brightness of pixels in the tracked region stays the same as the average brightness of pixels in the template.

(c) For the second method, we used Huber's Loss instead of least squares. That does not let outliers adversely affect the cost function evaluation. This turns our previous problem from least square problem to weighted least square problem, i.e. for each residual term you will have a corresponding weight ii and your minimization function will look like

$$L = \sum_x \wedge_{ii}[T\left(W\left(x;\Delta p\right)\right) - I\left(W\left(x;p\right)\right]^2$$

$$\Delta p = (A^T \wedge A)^{-1} A^T \wedge b$$

(d) Here $\wedge$ is a diagonal matrix of weights corresponding to the residual term computed as per the choice of the robust M-estimator used, A is the Jacobian matrix for each pixel of the template considered in the cost function, and b is the corresponding vector of residuals.

7.2.1 Robustness: Scaling Intensities

(a) To make our algorithm robust to illumination, we have calculated the mean of the template frame and the current frame.

(b) We then use the following formula to get the intensities of the current frame.

$$frame_{\ current} = \frac{Mean\ of\ template\ intensities}{mean\ of\ current\ frame\ intensities} * frame_{\ current}$$

(c) In this case we thus scale the intensities of the current frame according to the template frame then pass to LK and the LK can now track the template.

7.2.2 Robustness: Weighted Errors

(a) Instead of giving same weightage to all the errors we gave weights to the errors by which the areas with more error will be penalized more.

(b) And thus, the effect of outliers in the output will greatly reduce.

(c) We calculated the mean and variance of error and then if its in one standard deviation then it was given a higher value and lower value was given to the outliers.

(d) We had a great help from all these resources:
- Lecture notes and pdf.[2]
- Computer Vision, A Modern Approach [3]
- Lucas-Kanade 20 Years On: A Unifying Framework: Part 1[1]

(e) Following is the output of this technique.

8 Results

8.1 Car

Figure 2: Car on Road

8.2 Baby

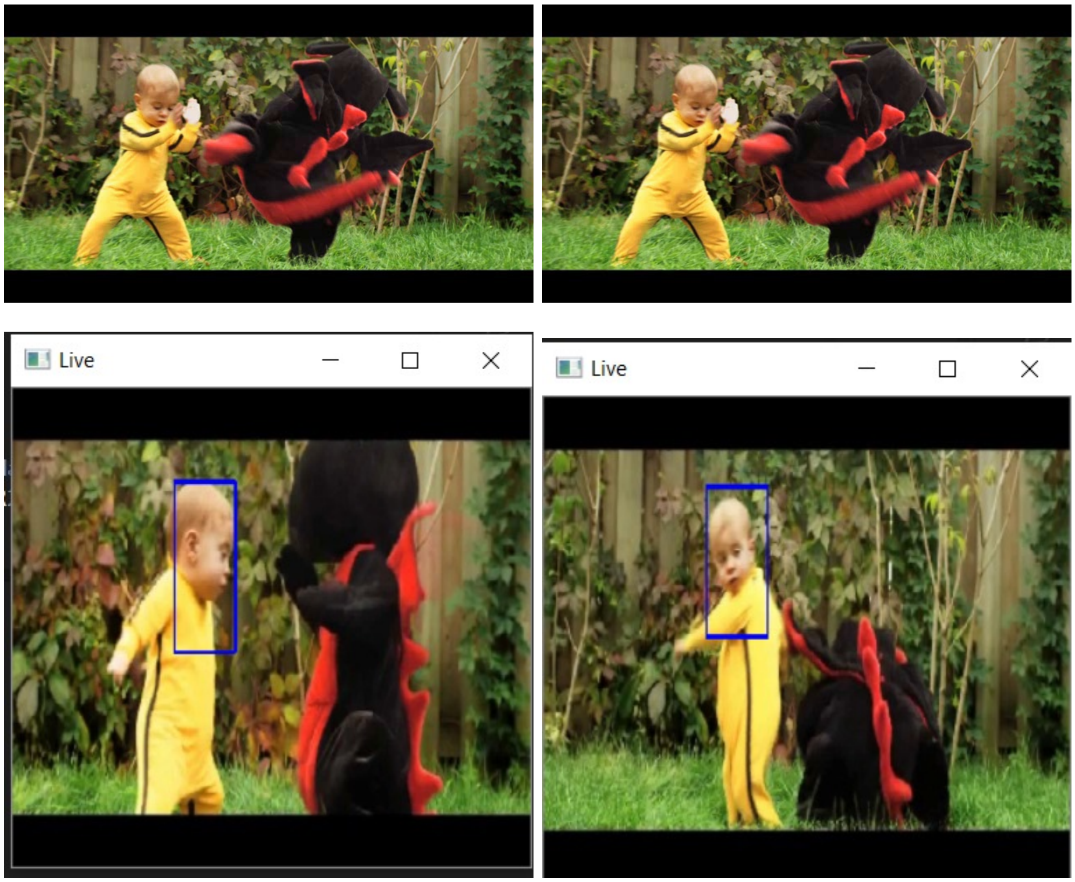

Figure 3: Human

8.3 Race

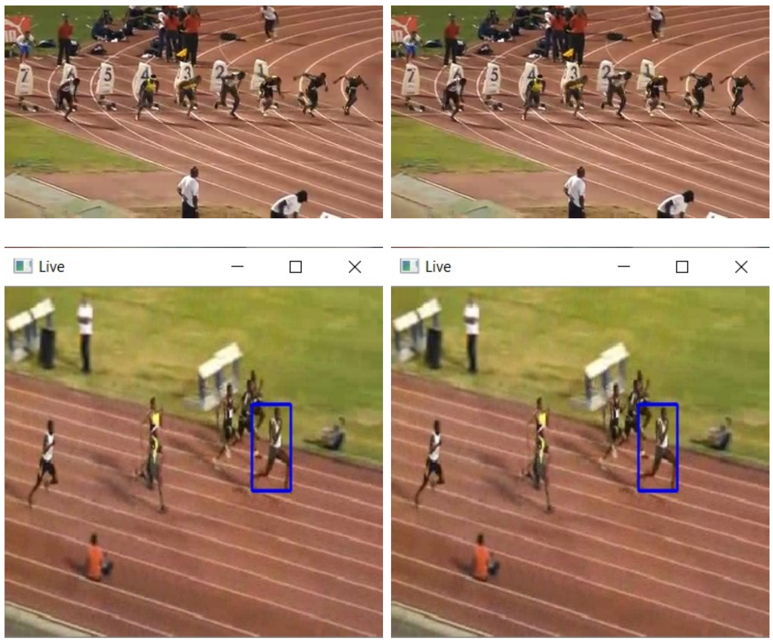

Figure 4: Usain Bolt's Race

References

[1] Simon Baker and Iain Matthews. *EM and Gaussian Mixture Models.*

- Research Paper on Lucas Kanade: A unifying framework.

.

[2] Prof. Charifa. *ENPM-673.*

- Lecture notes and pdf.

.

[3] Forsyth and Ponce. *Computer Vision, A Modern Approach.*

- Pdf - Computer Vision, A Modern Approach.

.