Project 4 Report

ENPM 673 - Perception for Autonomous Robots

Team 5

Prof. Dr. Mohammed Charifa

UNIVERSITY OF MARYLAND

Nikhil Lal Kolangara (116830768)
Kartik Venkat (116830751)
Kushagra Agrawal(116700191)

This paper represents our own work in accordance with University regulations.

Contents

1 Introduction

1.1 Summary

The goal of this project is to implement and build an object tracker which will be used on three given video sequences to evaluate our implementation. The three video sequences are from the Visual Tracker Benchmark Database: featuring a car on the road, a baby fighting a dragon, and Usain bolt's race. In order to start with object tracking, it is necessary to learn about the concept of Optical Flow.

1.2 Optical Flow

Optical flow also known as optic flow is defined as the pattern of apparent motion of any object in a given frame, surfaces, and edges in a given visual scene or a video sequence due to the relative motion observed between the scene and the point of view.[1]

It can be also defined as the pattern of apparent motion of image objects between two consecutive frames caused by the movement of object or the camera. It was introduced by American psychologist James J. Gibson in the 1940s to explain the visual stimulus provided to animals moving through the world.[2] He demonstrated its importance in affordance perception which is the ability to discern possibilities for action within the environment. [1]

The concept of optical flow has widely been adopted by roboticists, involving techniques that are related to image processing and control of navigation that are motion detection, object segmentation, time-to-contact information, focus of expansion calculations, etc. [3][4] The major use of optical flow has been inn the field of robotics where the researchers work on it to implement object detection and tracking, image dominant plane extraction, determining direction of movement for robot navigation and visual odometry.[3]

Optical flow analysis has few assumptions:

1. The pixel intensities of an object do not change between consecutive frames. 2. Neighbouring pixels have similar motion.

There are various optical flow determinataion function templates. The one we are using in this project is Lucas-Kanade template.

2 Lucas-Kanade

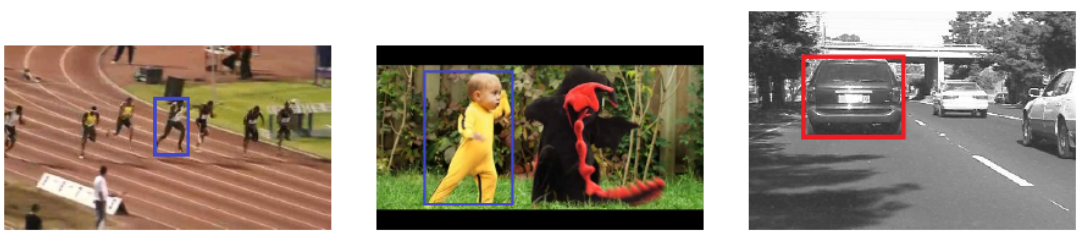
Lukas-Kanade method is a type of differential optical flow estimation method in which it assumes that given a point in a frame the optical flow is basically tracking the point as well as its neighboring pixels in the next frame meaning the flow is constant in a local neighbourhood of that point and the optical flow equations are solved for all of its neighbourhood pixels by least square criterion.[5][6]

The inherent ambiguity present in the optical flow equation is resolved by this method by combining the information from several neighbourhood pixels. There lies a disadvantage in Lucas-Kanade method as it cannot provide the flow information of any interior of uniform regions inside the image since it is purely a local method whereas the advantage is that it is also less susceptible to image noise.

3 Implementation

Given 1 *In this project you will implement the Lucas-Kanade (LK) template tracker. Then you will evaluate your 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. The short video that you will process are available in the data section. To initialize the tracker you need to define a template by drawing a bounding box around the object to be tracked in the first frame of the video. For each of the subsequent frames the tracker will update an affine transform that warps the current frame so that the template in the first frame is aligned with the warped current frame. For extra credit, you may look at ways to make tracking more robust.*

Evaluate your tracker on the three sequences: the car sequence, the bolt sequence, and the baby fighting a dragon. Use only the grayscale, not the color. What sort of templates work well for tracking? At what point does the tracker break down? Why does this happen?



(a) Bolt (b) Car (c) DragonBaby

Figure 1: Tracking Sequences

Proof 1 *As discussed previously, the optical flow methods involves taking in a given set of points and a frame. Later, it will attempt to find those points in the next frame. The point to track and its neighbourhood points are given by us.*

Step 1: Object Detection to detect the object - For the object detection part, a point is selected in the region of interest, and the neighbouring points are selected by using a bounding box with different colors.

Step 2:The next part is to use optical flow to show the flow of this selected object.

Lucas-Kanade template is used to track the optical flow. The goal of Lucas-Kanade is to align a template image T(x) to an input image I(x), where $x = (x, y)^T$ *is a column vector containing the pixel coordinates.[5]*

NOTE: It is impossible to determine whether the camera is moving or the object is moving in the video sequence and hence we assume based on the sequence and the Lucas-Kanade optical flow tracker works best for both the scenarios.

Step 3: We create the tracker function(discussed below) which takes in the previous frame, previous points and the current frame. The basic ideas is to detect the point to track, take the previous frame and look at the current frame and locate the points from present frame and find them in the next frame.

Step 4: The goal of the Lucas-Kanade algorithm is to minimize the sum of squared error between two images, the template T and the image I warped back onto the coordinate frame of the template: $\sum_x [I(W(x;p)) - T(x)]^2$ *Warping I back to compute I(W(x; p)) requires interpolating the image I at the sub-pixel locations W(x; p). [5]*

Step 5: The algorithm is as follows - All the steps are iterated until the estimates of the parameters of p converge.[5]

*Step 6: Warp I with **W(x;p)** to compute I(**W(x;p)**). Compute the error image T(x) - I(**W(x;p)**)*

Step 7: Warp the gradient ΔI *with W(x;p) and evaluate the Jacobian* $\frac{\partial W}{\partial p}$ *at (x;p).*

Step 8: Calculate the steepest descent images $\Delta I \frac{\partial W}{\partial p}$ *after which calculate the Hessian Matrix using the following equation* $H = \sum_x [\Delta I \frac{\partial W}{\partial p}]^T [\Delta I \frac{\partial W}{\partial p}$

Step 9: Compute $\sum_x [\Delta I \frac{\partial W}{\partial p}]^T [T(x) - I(W(x;p))]$ *and calculate* Δp *using the following equation*

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

Step 10: Finally, update the parameters p ¡- p + Δp

4 Evaluation of the Tracker

1. During our implementation it was observed that the objects to be tracked are moving a lot, so for that tight bounding boxes were required. As the background creates noises, the least portion of the background were required.

2. In case of Bolt video sequence example, initially the head of all the athletes are down but later in the video the head of Bolt straightens up, but of the other athletes beside him is still down which fails the algorithm and a different athlete is being tracked.

3. The tracker fails when the object is moving too fast or when there is change in illumination. In case of fast moving objects the optical flow vector becomes too large and hence the Lucas-Kanade tracker fails. In case of illumination changes the algorithm fails because the sum of square distance errors that it tries to minimize is sensitive to illumination changes.

5 Robustness to Illumination

1. To make our algorithm robust to illumination we have taken the mean of the intensities of the template and the current frame.

2. The following formula is used to calculate the intensity:

$$frame_{current} = \frac{meanoftemplateintensities}{meanofthecurrentframeintensities} * frame_{current}$$

3. The intensity of the current frame with respect to the template frame and then it is passed to the Lucas-Kanade tracker function.

4. In case of weighted error, same weight is not given to all the errors instead the weight is given to those error by which the areas with more error will be penalized more, thus the effect of outliers will greatly reduce.

5. Following is the formula for this technique:

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

6. Centeroid is calculated from the coordinates which we get after the warping is done and that centeroid is used to draw the bounding box as the bounding box drawn after the warping is done goes out of bounds for some frames.

6 Output

Car Tracking:

Figure 2: Car in the video sequence

Figure 3: Template of the Car

Figure 4: Tracking the Car

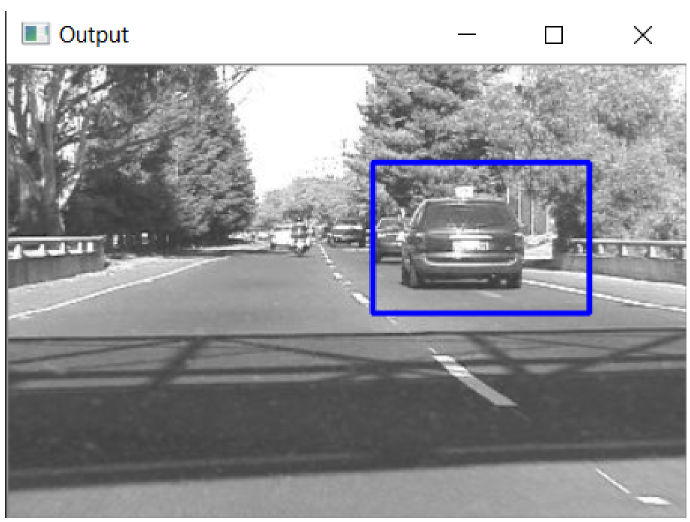


Figure 5: Weighted Error Method

Baby Tracking:

Figure 6: Baby in the video sequence

Figure 7: Template of the Baby

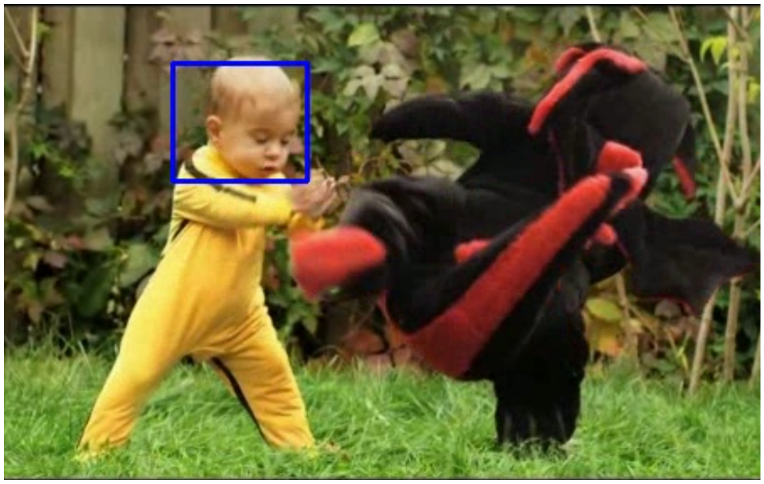


Figure 8: Tracking the Baby

Bolt Tracking:

Figure 9: Usian Bolt in the video sequence

Figure 10: Template of Usain Bolt

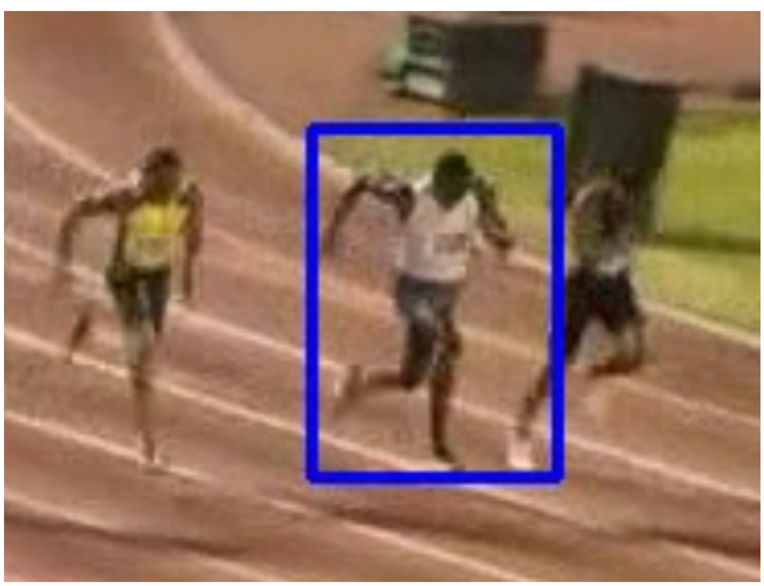


Figure 11: Tracking Usain Bolt

Drive Link:

https://drive.google.com/drive/folders/1NDMPIL46TM7aRbKyKFNos0w1mSbgxd-G?usp=sharing

7 References

[1] https://en.wikipedia.org/wiki/Optical_flow

[2] Gibson, J.J. (1950). The Perception of the Visual World. Houghton Mifflin.

[3] Aires, Kelson R. T.; Santana, Andre M.; Medeiros, Adelardo A. D. (2008). Optical Flow Using Color Information (PDF). ACM New York, NY, USA. ISBN 978-1-59593-753-7.

[4] Beauchemin, S. S.; Barron, J. L. (1995). The computation of optical flow. ACM New York, USA.

[5] B. D. Lucas and T. Kanade (1981), An iterative image registration technique with an application to stereo vision. Proceedings of Imaging Understanding Workshop, pages 121–130

[6]Bruce D. Lucas (1984) Generalized Image Matching by the Method of Differences