Object Tracking Using Optical Flow With Kalman Filter

Valmiki Kothare
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
vok@andrew.cmu.edu

Michael F. Finch
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213
mfinch@andrew.cmu.edu

Tejas Srivatsav
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
tsrivats@andrew.cmu.edu

1 Introduction

Object tracking is an area of development that is constantly evolving as more practical applications for it arise. Obstacle detection/avoidance in autonomous vehicles, gesture tracking and hand movement for VR, and multi-object position/velocity calculations can all be implemented using object tracking. Many of these applications, however, require low latency response times in order to run effectively in real-time. At the same time, many naive object tracking algorithms are computationally intensive and are ill-suited for real-time applications. One such approach, which uses optical flow to estimate the velocity of pixels/objects in a video, is particularly expensive, but effective for determining positional and velocity data for objects in a video. In order to speed up computation, Kalman filters can be used to make predictions for the future state (i.e. position and velocity) of the system based on the current state. This estimate, which is computationally cheap relative to full object detection, can be used to boost performance of the object tracking algorithm, thus allowing it to be used in real-time applications. This work investigates the use of the Lucas-Kanade method of optical flow feature extraction to be used for object acquisition and tracking. The optical flow vector determined for the object velocity will then be used as an input state vector into a Kalman filter which will estimate the relative position and velocity of the object for future frames of the video.

The corresponding code can be found at: https://github.com/tsrivatsav/MLSP-Team-16

2 Related Work

Optical flow describes the motion of light intensity between image frames as a vector field. The analysis of this vector field is used to determine the motion of objects across the field of view. The two classic methods used for determining optical flow is the Lucas-Kanade and Horn-Schunck methods, where the optical flow equation is formulated from a Taylor series expansion approximation of pixel brightness change in the vertical and horizontal directions of each consecutive frame[1,2]. Applications for optical flow include motion estimation, video compression, object tracking, object detection, and navigation to name a few. The Kalman filters are used for state estimation or future state prediction given the measurement of the current state [3]. Specifically, when applied to object tracking, Kalman filters can be employed to improve estimation accuracy of object position and velocity.

Preprint. Under review.

3 Optical Flow Background

Optical flow uses the change of pixel intensity in both time and space between frames of a video to determine motion of an object or target of interest. The velocity for the target, assuming the target is slow moving, can be determined using the following equation:

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y = -\frac{\partial I}{\partial t}. \tag{1}$$

The optical flow equation is formulated from a linearized truncation Taylor series and can be written in matrix notation as follows:

$$\mathbf{A}\mathbf{v} = \mathbf{b}, \tag{2}$$

where

$$\mathbf{A} = \begin{bmatrix} I_x(q1) & I_y(q1) \\ I_x(q2) & I_y(q2) \end{bmatrix}, \tag{2a}$$

$$\mathbf{v} = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \tag{2b}$$

$$\mathbf{b} = \begin{bmatrix} -I_t(q1) \\ -I_t(q2) \end{bmatrix}. \tag{2c}$$

Eq. 2a and 2c assume a window function that only considers a neighborhood of four pixel in calculating the localized optical flow. Formulating the completed matrix, $\mathbf{A}$, for all pixels in the image and computing $\mathbf{A^T A}$ result in the structure tensor. The summation of the diagonal components, i.e. $|I_x|^2 + |I_y|^2$, results in the segmentation mask show in figs 1-3 (b). Finite difference approximation[4] is employed to calculated the gradient of the image with respect to space and time using the convolution kernel: $\mathbf{k} = \begin{bmatrix} -\frac{1}{2} & 0 & -\frac{1}{2} \end{bmatrix}$.

The localized estimation of the optical flow, or velocity, can be determined using the Lucas-Kanada method [2] by using the pseudo-inverse on eq. 2.

4 Kalman Filter Background

Kalman Filtering is a recursive prediction algorithm that uses a series of observations over time to estimate unknown variables. In this case, the variables that we want to estimate are the object's position and velocity in the x and y directions. The algorithm consists of 3 parts.

<u>Prediction</u>

$$X_t = AX_{t-1} + Bu \tag{3}$$

$$P_t = AP_{t-1}A^t + Q \tag{4}$$

To be clear, X represents the state of the object and is a size 4 vector that contains its x and y positions and velocities. P represents the Covariance matrix and is of size 4x4. u contains the velocity and acceleration of the object - the latter of which we assumed to be zero. Q represents noise. And lastly, A and B are matrices that contain the appropriate kinematics equations to transform X over time.

<u>Measurement</u>

$$Y = Z - HX_t \tag{5}$$

$$K = P_t H^t / ((HP_t H^t) + R) \tag{6}$$

Here Z is a size 2 vector that contains the actual measurements being received. H is a 2x4 matrix used to convert X to a size that is compatible with Z. And lastly, R is a 2x2 matrix representing the uncertainty in our measurement. All of this is used to calculate the Kalman gain K - which represents the percent of uncertainty in our prediction as compared to the total uncertainty (including uncertainty in our measurement).

Update

$$X_t = X_t + KY \tag{7}$$

$$P_t = (I - KH)P_t \tag{8}$$

Finally, we use the difference in our predicted value and measurement along with the level of uncertainty in order to update the state vector and the correponding covariance matrix.

5 Dataset

In order to objectively test the validity of our optical flow algorithm and determine the performance boost from the Kalman filter, we constructed a dataset consisting of a square roaming randomly across a static background, with position and velocity recorded to validate our object detection and tracking results. We chose to use a trivial moving object example as opposed to practical data from video recordings because we hoped show the performance benefit from using a Kalman filter in conjunction with an object tracking algorithm based on optical flow, thus the added complexities of object recognition involved in using a more complex dataset was deemed unnecessary.

6 Results

Figs 1-3 show the resulting localized optical flow calculation on three frames of a video of a ball bouncing off of a planar surface. The optical flow calculation results in the isolation of the moving ball target and region of interested for the stationary planar surface. As is to be expected, the optical flow calculation shows a zero velocity when the ball decelerates and deforms on the planar surface.

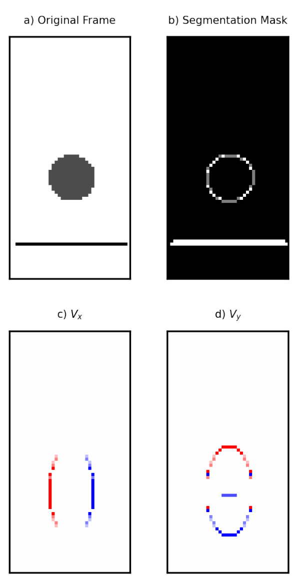


Figure 1: a) Original Frame b) segmented object and velocity (optical flow) in the x (c) and y (d) directions for target directed downward

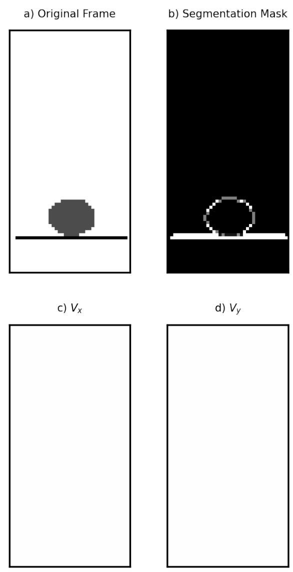


Figure 2: a) Original Frame b) segmented object and velocity (optical flow) in the x (c) and y (d) directions for target at zero velocity

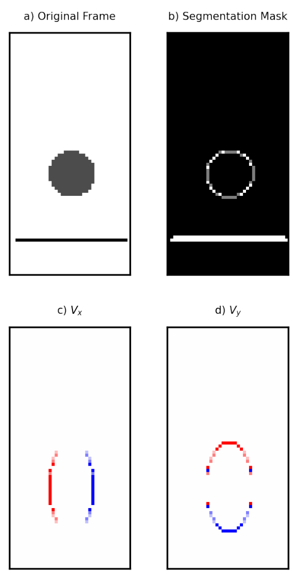
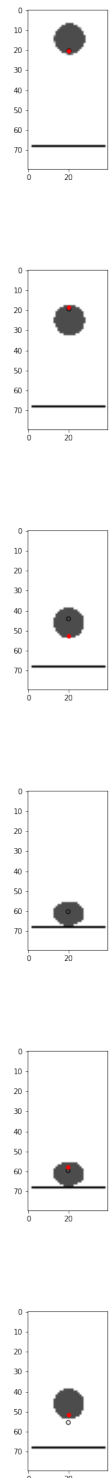
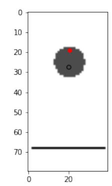


Figure 3: a) Original Frame b) segmented object and velocity (optical flow) in the x (c) and y (d) directions for target directed upward

Figure 4 shows the addition of the Kalman filter on eight frames of a video of a ball bouncing off of a planar surface. The output of the optical flow is indicated by the red dot and the prediction given by the Kalman filter is indicated by the black circle. It is interesting to note that the Kalman filter predicts the ball to be slightly behind where it is due to our assumption that objects will have zero acceleration.





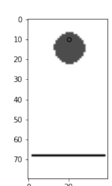


Figure 4: The ball bounces down and then up. The red dot is the output of optical flow and the black circle is the Kalman filter prediction.

Figures 5-9 shows the application of optical flow and Kalman filter to five frames of a video in which a person moves from the right side of the image to the left side. For the colored images, the optical flow output is indicated by the red dot and the prediction given by the Kalman filter is indicated by the blue circle. The gray-scale images also represent the output of the optical flow algorithm, as the intensity (whiteness) of each pixel corresponds to the magnitude of its relative velocity compared to the rest of the image. In this case, the pixels surrounding the person are white because the person is moving relative to the background.

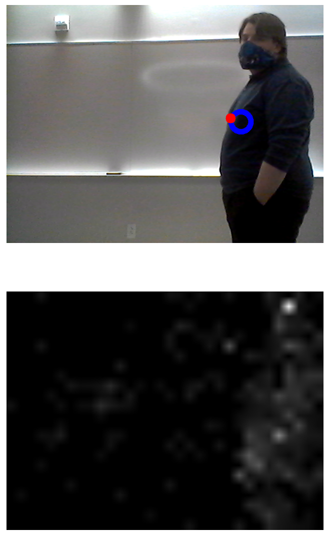


Figure 5: The person is on the right side of the image.

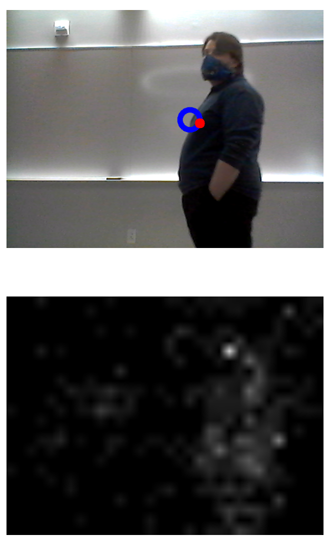


Figure 6: The person is between the middle and right side of the image.

Figure 7: The person is in the middle of the image.

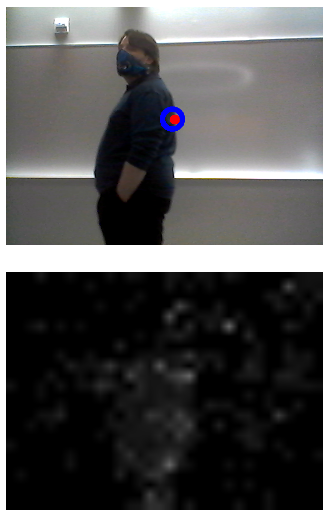


Figure 8: The person is between the middle and left side of the image.

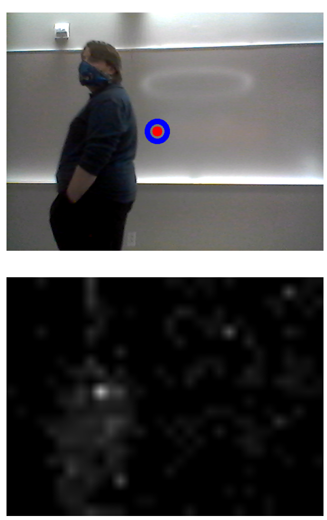


Figure 9: The person is on the left side of the image.

7 Discussion and Analysis

From experimentation it has been very clear that doing the optical flow calculation for every frame is simply too computationally expensive. Doing so results in a stuttered form of tracking that catches up to the object every few seconds instead of tracking it smoothly and consistently. Consequently, we used the Kalman filter every frame in order to predict where the object will be and computed the optical flow every five frames. We found five frames to be the ideal parameter for this exercise as it enabled the algorithm to maintain smoothness without sacrificing too much accuracy. Doing the optical flow calculation more sporadically would result in a decrease in accuracy whereas doing so more frequently would result in a decrease in smoothness.

8 Future Considerations

The addition of a Kalman filter to optical flow-based object tracking is mainly beneficial in domains where low latency is desirable. The Kalman filter allows for a drastic reduction in latency with negligible impact on accuracy. Potential applications include real-time problems such as hand tracking for VR devices, as well as obstacle detection and prediction for autonomous vehicles.

Future extensions of this work could include detection and tracking for multiple objects within one image. This can be done by deconstructing the image into several parts and running the algorithm on each of those parts. Additionally, the same calculation can be extended to account for non-zero acceleration. Although this would lead to larger matrices and thus more computation, the general ideas and equations would be virtually identical and therefore easy to implement.

References

[1] Pan, J.N. & Shi, Y.Q. (1994) A Kalman filter for improving optical flow accuracy along moving boundaries *Proc. SPIE 2308, Visual Communications and Image Processing '94* **2308**:638-649.

[2] Oskoei, M.A. (2017) Adaptive Kalman filter applied to vision based head gesture tracking for playing video games *Robotics* **6**(4):33.

[3] Song, X., Seneviratne, L. D. & Althoefer, K. (2010) A Kalman filter-integrated optical flow method for velocity sensing of mobile robots *IEEE/ASME Transactions on Mechatronics* **16**(3):551-563.

[4] Sadiku, M.N. (2009) *Numerical Techniques in Electromagnetic with MATLAB*. CRC Press.