Lucas Kanade Template Tracker

Arjun Srinivasan Ambalam, Praveen Menaka Sekar, Arun Kumar Dhandayuthabani

1 Lucas-Kanade Tracker

1.1 Introduction

Optical flow is a family of techniques used to estimate the pattern of the apparent motion of image objects caused by the movement of objects or a camera in a video sequence. It is a vector field where each vector has two components to show the displacement between points due to their movement from the first image (also called frame) to the second image. This algorithm assumes that the flow is essentially constant in the local neighborhood of the pixel under consideration, and solves the basic optical flow equations for all of the pixels in that neighborhood with the least squares criterion. In this project we implement two variants of the Lucas-Kanade template tracker, the Forward-Additive Lucas-Kanade template tracker and Inverse Compositional Lucas-Kanade template tracker on three video sequences featuring a car on the road, a baby fighting a dragon, and Usain bolt's race.

1.2 Forward-Additive Lucas-Kanade Tracker

The main aim of the Lucas-Kanade tracker is to align a template image $\mathbf{T(X)}$ to an input image $\mathbf{I(X)}$. The warp $\mathbf{W(X;\ p)}$, where $\mathbf{p} = (\mathrm{p_1},\ \mathrm{p_2},\ \mathrm{p_3},\ \mathrm{p_4},\ \mathrm{p_5},\ \mathrm{p_6})^{\mathrm{T}}$ is a vector of parameters, takes the pixel $\mathbf{X}$ in the coordinate frame of the template $\mathbf{T}$ and maps it to the sub-pixel location $\mathbf{W(X;\ p)}$ in the coordinate frame of image $\mathbf{I}$.

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

Now, the next important aspect of Lucas-Kanade tracker is to minimize the sum of squared error between the template ($\mathbf{T}$) and the image ($\mathbf{I}$) warped back onto the coordinate frame of the template ($\mathbf{T}$) with respect to the parameters ($\mathbf{p}$).

$$min(\sum_x [\mathbf{I(W(X;\ p))} - \mathbf{T(X)}]^2) \tag{2}$$

To optimize the expression in above equation, the Lucas-Kanade algorithm assumes that we know the current estimate of $\mathbf{p}$ and then iteratively solve for the increments to the parameters.

$$min(\sum_x [\mathbf{I(W(X;\ p\ +\ \Delta p))} - \mathbf{T(X)}]^2) \tag{3}$$

We optimize the above equation with respect to Δp and update the parameters $\mathbf{p} = \mathbf{p} + \Delta p$. The above two steps are performed until parameter estimates $\mathbf{p}$ converge i.e. $\|\mathbf{p}\| \leq \epsilon$.

The above non-linear equation can be linearized using first order Taylor equation which is shown below,

$$min(\sum_x [\mathbf{I(W(X;\ p\ +\ \Delta p))} + \Delta\ \mathbf{I} \frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}} \Delta\ \mathbf{p} - \mathbf{T(X)}]^2) \tag{4}$$

Where,

- $\Delta\ \mathbf{I}$ - gradient of image $\mathbf{I}$ computed at $\mathbf{W(X;\ p)}$.

- $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ - Jacobian of the warp,

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{bmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{bmatrix} \tag{5}$$

The solution for equation (4) can thus be obtained by taking a partial derivative with respect to $\Delta\ \mathbf{p}$ and equating it to 0. Therefore the optimal solution is given by,

$$\Delta\ \mathbf{p} = H^{-1} \sum_x [\Delta\ \mathbf{I} \frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}]^{\mathrm{T}} [\mathbf{T(X)} - \mathbf{I(W(X;\ p))}] \tag{6}$$

Where,

- $\Delta\ \mathbf{I} \frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}$ - steepest descent images
- H - n×n Hessian matrix

$$H = \sum_x [\Delta\ \mathbf{I} \frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}]^{\mathrm{T}} [\Delta\ \mathbf{I} \frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}] \tag{7}$$

Using the above obtained update parameter we evaluate the new parameter vector and then iteratively solve until convergence. The implementation of the Forward-Additive Lucas-Kanade tracker based on the above given mathematical background is shown below.

Algorithm 1: Forward-Additive Lucas-Kanade

Result: updated parameters $\mathbf{p}$

Iterate till $\|\mathbf{p}\| \leq \epsilon$;

1. Warp $\mathbf{I}$ with $\mathbf{W(X;\ p)}$ to compute $\mathbf{I(W(X;\ p))}$
2. Compute error image $\mathbf{T(X)}$ - $\mathbf{I(W(X;\ p))}$
3. Warp the gradient $\Delta\ \mathbf{I}$ with $\mathbf{W(X;\ p)}$
4. Compute Jacobian $\frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}$ at (X; p)
5. Compute steepest descent images $\Delta\ \mathbf{I} \frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}$
6. Compute the hessian matrix H = $\sum_x [\Delta\ \mathbf{I} \frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}]^{\mathrm{T}} [\Delta\ \mathbf{I} \frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}]$
7. compute $\Delta p = H^{-1} \sum_x [\Delta\ \mathbf{I} \frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}]^{\mathrm{T}} [\mathbf{T(X)} - \mathbf{I(W(X;\ p))}]$
8. Update the parameters $\mathbf{p} \leftarrow \mathbf{p} + \Delta p$

1.3 Inverse Compositional Lucas-Kanade Tracker

While implementing the Forward-Additive Lucas-Kanade tracker, the major problem we faced was its speed. Calculating the affine transform and Hessian for every iteration, just for a single image is computationally heavy. In order to overcome this issue we chose to implement the Inverse Compositional Lucas-Kanade algorithm. Inverse Compositional Lucas-Kanade algorithm achieves this by interchanging the use of image ($\mathbf{I}$) and template ($\mathbf{T}$). The aim of Inverse Compositional Lucas-Kanade algorithm is still to minimize the squared error between image ($\mathbf{I}$) and template ($\mathbf{T}$), only difference being that the places of image ($\mathbf{I}$) and template ($\mathbf{T}$) are interchanged.

$$min(\sum_x [\mathbf{T(W(X;\ p\ +\ \Delta p))} - \mathbf{I(W(X;\ p))}]^2) \tag{8}$$

We optimize the above equation with respect to Δp and update the warp,

$$\mathbf{W(X;\ p)} = \mathbf{W(X;\ p)} \circ \mathbf{W(X;\ p\ +\ \Delta p)}^{-1} \tag{9}$$

with the parameter updates given by,

$$\frac{1}{(1+p_1)\cdot(1+p_4)-p_2\cdot p_3}\begin{bmatrix} -p_1-p_1\cdot p_4+p_2\cdot p_3 \\ -p_2 \\ -p_3 \\ -p_4-p_1\cdot p_4+p_2\cdot p_3 \\ -p_5-p_5\cdot p_4+p_6\cdot p_3 \\ -p_6-p_1\cdot p_6+p_2\cdot p_5 \end{bmatrix} \tag{10}$$

All affine warps with $(1 + p_1)\cdot (1 + p_4)$ - $p_2\cdot p_3 = 0$ are omitted. The above two steps are performed until parameter estimates $\mathbf{p}$ converge i.e. $\|\mathbf{p}\| \leq \epsilon$. The above non-linear equation can be linearized using first order Taylor equation which is shown below,

$$min(\sum_x [\mathbf{T(W(X;\ 0)} + \Delta\ \mathbf{T}\frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}\Delta\ \mathbf{p} - \mathbf{I(W(X;\ p)}]^2) \tag{11}$$

Where,

- $\Delta\ \mathbf{T}$ - gradient of image $\mathbf{T}$ computed at identity warp $\mathbf{W(X;\ 0)}$.
- $\frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}$ - Jacobian of the warp,

$$\frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}} = \begin{bmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{bmatrix} \tag{12}$$

The solution for equation (11) can thus be obtained by taking a partial derivative with respect to $\Delta\ \mathbf{p}$ and equating it to 0. Therefore the optimal solution is given by,

$$\Delta\ \mathbf{p} = H^{-1}\sum_x [\Delta\ \mathbf{T}\frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}]^{\mathrm{T}}[\mathbf{I(W(X;\ p))} - \mathbf{T(X)}] \tag{13}$$

Where,

- $\Delta\ \mathbf{T}\frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}$ - steepest descent images
- H - n×n Hessian matrix

$$H = \sum_x [\Delta\ \mathbf{T}\frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}]^{\mathrm{T}}[\Delta\ \mathbf{T}\frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}] \tag{14}$$

Using the above obtained update parameter we evaluate the new parameter vector and then iteratively solve until convergence. The implementation of the Inverse Compositional Lucas-Kanade tracker based on the above given mathematical background is shown below.

Algorithm 2: Inverse Compositional Lucas-Kanade

Result: updated warp $\mathbf{W(X;\ p)}$

Pre-compute:

1. Compute gradient of the template $\mathbf{T(X)}\ \Delta\ \mathbf{T}$
2. Compute Jacobian $\frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}$ at (X; 0)
3. Compute steepest descent images $\Delta\ \mathbf{I}\frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}$
4. Compute steepest descent images $\Delta\ \mathbf{T}\frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}$

Iterate till $\|\mathbf{p}\| \leq \epsilon$;

1. Warp $\mathbf{I}$ with $\mathbf{W(X;\ p)}$ to compute $\mathbf{I(W(X;\ p))}$
2. Compute error image $\mathbf{I(W(X;\ p))}$ - $\mathbf{T(X)}$
3. compute $\Delta p = H^{-1} \sum_x [\Delta\ \mathbf{T}\frac{\partial\ \mathbf{W}}{\partial\ \mathbf{p}}]^{\mathrm{T}}[\mathbf{I(W(X;\ p))} - \mathbf{T(X)}]$
4. Update warp $\mathbf{W(X;\ p)} \leftarrow \mathbf{W(X;\ p)} \circ \mathbf{W(X;\ p\ +\ \Delta p)}^{-1}$

1.4 Evaluation of the Tracker

- Lucas-Kanade algorithm implies a model of expected spatial transformations between template and observed image. Therefore, as long as a template maintains a smooth and almost constant or linear spatial relationship with the image, it works well.
- Partial occlusion, rapid change in illumination, and orientation changes in the scene all cause the Lucas-Kanade tracker to break down due to it's heavy dependency on spatial relationship between pixels.
- Mathematically, the above scenarios lead to a degenerate affine warps, thereby we end up with a singular Hessian matrix and hence the tracker no longer responds.

1.5 Robustness to Illumination

- Our first approach to enhance the performance of our tracker to illumination and other appearance changes in the scene was to update our template frame to frame.
- Though this enhances our tracker performance to some extent for the bolt video sequence, the tracker still performed poorly for both car and baby video sequence.
- In case of the car video sequence, the illumination changes are so rapid under the bridge that updating our template frame to frame is not sufficient.
- On the other hand, distinction between the background and tracked object is so ambiguous that updating the tracker worsens tracking even more. For example, if the baby's head turned and the template is updated it is possible that the background would be incorporated and pushes the template away from the head.
- In order to enhance the performance of tracker for car video sequence, we implemented Contrast Limited Adaptive Histogram Equalization (CLAHE), Gamma Correction and mean-shift tracking.
  - We perform CLAHE on the warped image using cv2.createCLAHE inorder to take care of the image contrast .

– On the other hand, Gamma correction controls the overall brightness of an image.

$$I' = 255(\frac{I}{255})^{\frac{1}{\gamma}} \tag{15}$$

– Also, we tracked the shift in pixel mean with the help of Z score.

$$\sigma_{\mathrm{i}} = \frac{x_{\mathrm{i}} - \bar{x}_{\mathrm{i}}}{N} \tag{16}$$

$$z_{\mathrm{i}} = \frac{x_{\mathrm{i}} - \bar{x}_{\mathrm{t}}}{\sigma_{\mathrm{i}}} \tag{17}$$

$$x_{\mathrm{new,\ i}} = \bar{x}_{\mathrm{i}} + z_{\mathrm{i}}\sigma_{\mathrm{i}} \tag{18}$$

- In order to enhance the performance of tracker for baby video sequence, we implemented background subtraction along with Inverse Compositional Lucas-Kanade algorithm.
- Background Subtraction detects movement or significant differences inside of the frame, and removes all the non-significant components.
- We have also implemented weighted least-squares estimator i.e Huber Loss and found significant improvement in the performance of the tracker.

$$min(\sum_{x} A_{\mathrm{ii}}[\mathbf{T(W(X;\ p\ +\ \Delta p))} - \mathbf{I(W(X;\ p))}]^2) \tag{19}$$

Where, A_{ii} - weights corresponding to the residual term

1.6 Results

As mentioned before, the Inverse Compositional Lucas-Kanade algorithm performs better than Forward-Additive Lucas-Kanade in terms of speed. Hence, we chose to use it for our final results. The templates we used to implement the tracker can be seen below.

Figure 1: Bolt Template; Car Template; Baby Template

The tracking results using the above template can be seen below.

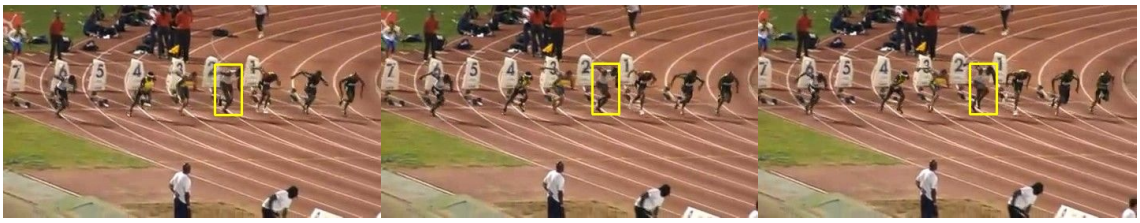

Figure 2: LK tracker result for Bolt video sequence

Figure 3: LK tracker result for Car video sequence

The output videos pertaining to the project can be found in the below link

- Lucas-Kanade Tracker Output videos

References

[1] Simon Baker, and Iain Matthews. ***Lucas-Kanade 20 Years On: A Unifying Framework: Part 1***. The Robotics Institute, Carnegie Mellon University.

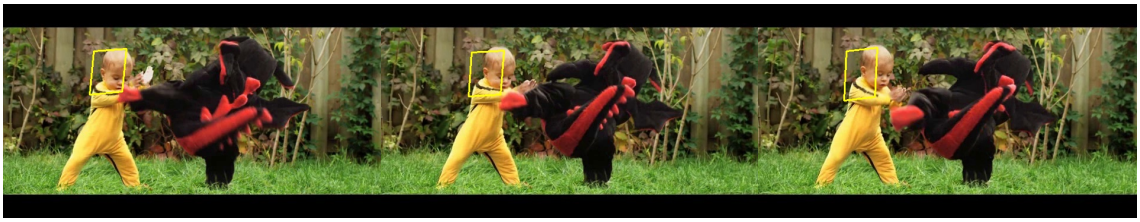

Figure 4: LK tracker result for Baby with Dragon video sequence