Computer Vision (COL780)

Assignment-2 Report

K N Ajay Shastry(2021CSY7547)

Suraj Patni (2021CSY7550)

1. **Appearance-based Techniques**

- Below is the comparison of the Block-Based algorithm using SSD and NCC for all our datasets. (max IOU is marked in blue color)

DataSet	**IOU (SSD)**	**IOU (NCC)**
Blur Car	0.3356	0.537
Bolt	0.006	0.0049
Liquor	0.6	0.52

Problems with Appearance-based matching techniques

- We can't perform affine and projective transformations in Block based matching approach. So this algorithm is not able to perform warps that involve affine transformations or homography.
- In this method, we are not supposed to change our template image, so we can't handle most of the warps on the image. But in Lucas Kanade, we can make updates to the template image.

- This algorithm is not Robust because it is intensity-based, so it often detects the **wrong template**.
  For example: below are the output for Bolt Dataset.

2. Lucas Kanade Algorithm

Steps to track a template using Lucas Kanade:

1. Warp $I(\mathbf{x})$ with $W(\mathbf{x}, \mathbf{p})$ ---> $I(W(\mathbf{x}, \mathbf{p}))$
2. Compute the error: subtract $I(W(\mathbf{x}, \mathbf{p}))$ from $T(\mathbf{x})$ ($T(\mathbf{x})$ is our template)
3. Compute warped gradients: $\nabla I = [Ix, Iy]$, evaluated at $W(\mathbf{x}, \mathbf{p})$
4. Evaluate the Jacobian of the warping: $\partial W / \partial \mathbf{p}$
5. Compute the steepest descent: SD = ∇I *$(\partial W / \partial \mathbf{p})$
6. Compute Inverse Hessian:

$$H^{-1} = \left[\sum_{\mathbf{x} \in \mathbf{T}} \left[\nabla I \frac{\partial W}{\partial \mathbf{p}}\right]^{\mathrm{T}} \left[\nabla I \frac{\partial W}{\partial \mathbf{p}}\right]\right]^{-1}$$

7. Multiply the steepest descent with error:

$$\sum_{\mathbf{x} \in \mathbf{T}} \left[\nabla I \frac{\partial W}{\partial \mathbf{p}}\right]^{\mathrm{T}} \left[T(\mathbf{x}) - I(W(\mathbf{x}, \mathbf{p}))\right]$$

8. Compute $\Delta \mathbf{p}$
9. Update parameters: $\mathbf{p}$ ----> $\mathbf{p} + \Delta \mathbf{p}$
10. Repeat until $\Delta \mathbf{p} < \boldsymbol{\varepsilon}$

The resulting IOUs were as follows when we executed this algorithm using Affine Transforms on the data provided.

DataSet	**IOU**
Blur Car	0.3534
Bolt	0.16
Liquor	0.498

When we compare it with the results obtained in the block-based method, below are a few comparisons:

Result in the Block-Based method:

Result in the Lucas-Kanade method:

As seen above, the Lucas-Kanade method was able to track Usain Bolt with more precision than that of the Block-Based method. But in the other datasets, the Block-Based method outperformed the Lucas-Kanade method.

3. Pyramid based Lucas Kanade Algorithm

In this, we improvised on the standard Lucas Kanade algorithm. We used the method of scaling to improve the efficiency of the algorithm. Different datasets required different hyper-parameter tuning. Below are the results.

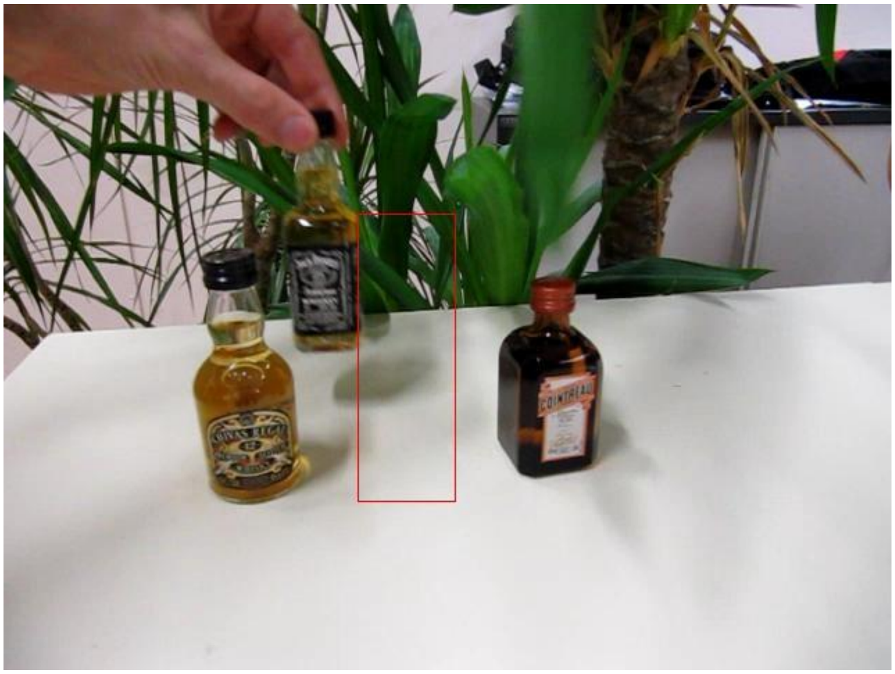
Data Set	IOU	Number of layers
Blur Car	0.44	3
Bolt	0.152	1
Liquor	0.552	2

Below are a few examples that demonstrate the improvement

The first pair of examples demonstrate the comparison between standard and pyramid-based Lucas Kanade for the Liquor Dataset.

The second pair of examples demonstrate the comparison between standard and pyramid-based Lucas Kanade for the Blur Car dataset.

Standard Lucas Kanade

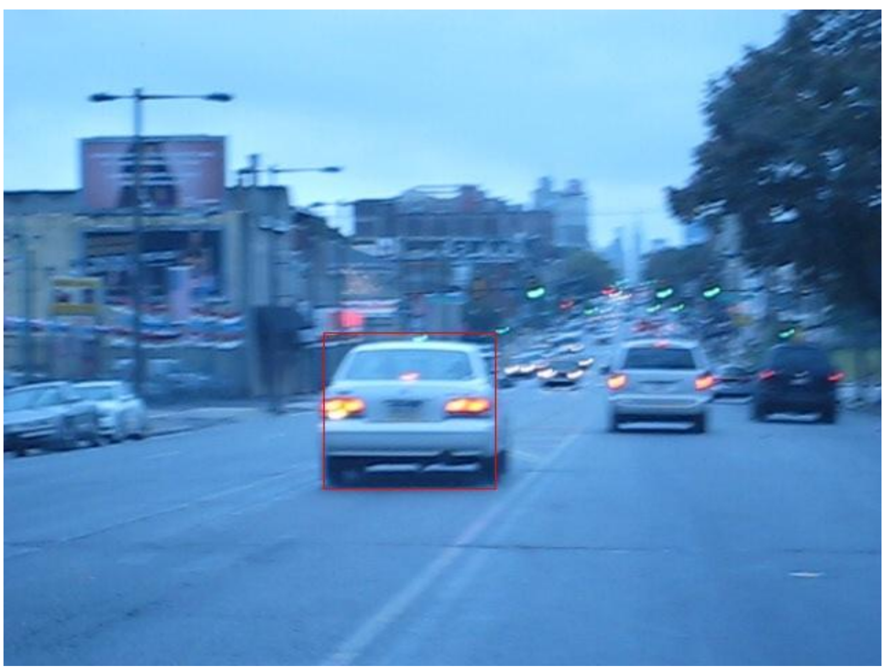


Pyramid Based Lucas Kanade

We see that Pyramid based method tracks the bottle in a better way.

Standard Lucas Kanade.

Pyramid Based Lucas Kanade.



We see that Pyramid based method tracks the car in a better way.

4. Live Demo

In this case, we have implemented the object tracking algorithm using the block-based template tracking method. The Lucas Kanade took a lot of time to process data in the live stream.

The results can be seen during the live demonstration.