1 ...

2 Theory Questions

2.1 Calculating the Jacobian

W(p) = $\begin{bmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \\ 0 & 0 & 1 \end{bmatrix}$

$J = \frac{\partial W}{\partial p} = \begin{bmatrix} \frac{\partial W}{\partial p_1} & \frac{\partial W}{\partial p_2} & \frac{\partial W}{\partial p_3} & \frac{\partial W}{\partial p_4} & \frac{\partial W}{\partial p_5} & \frac{\partial W}{\partial p_6} \end{bmatrix}$

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

2.2 Computational Complexity

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

initialization complexity
$O(np)$
$J^T J \;\; >>> \;\; O(p^2 n)$
$H^{-1} \;\; >>> \;\; O(p^3)$
$O(np + p^2 n + p^3)$

$O(p^2 n + p^3)$

each iteration complexity
$I(W(x;p)) - T \;\; >>> \;\; O(n)$
$J^T \;\; >>> \;\; O(np)$
$H^{-1} \;\; >>> \;\; O(p^2)$
$O(n + np + p^2)$

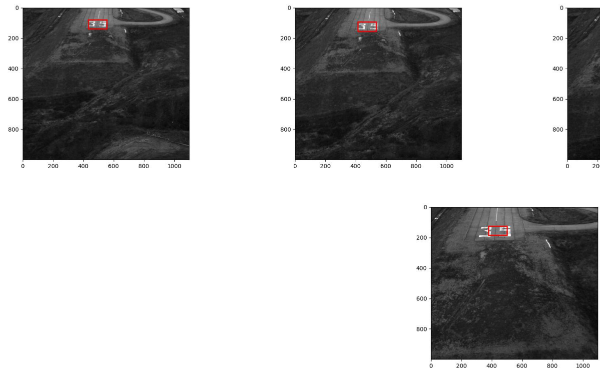
$O(np + p^2)$

Both the Lucas Kanade and the Matthews Baker methods have similar run time complexities. Both are mostly affected by the number of pixels in the template and the amount of warp parameters.

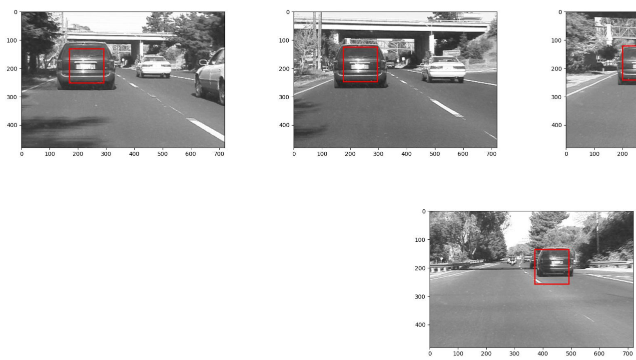
3 Lucas-Kanade Tracker

3.1 Lucas-Kanade Forward Additive Alignment with Translation

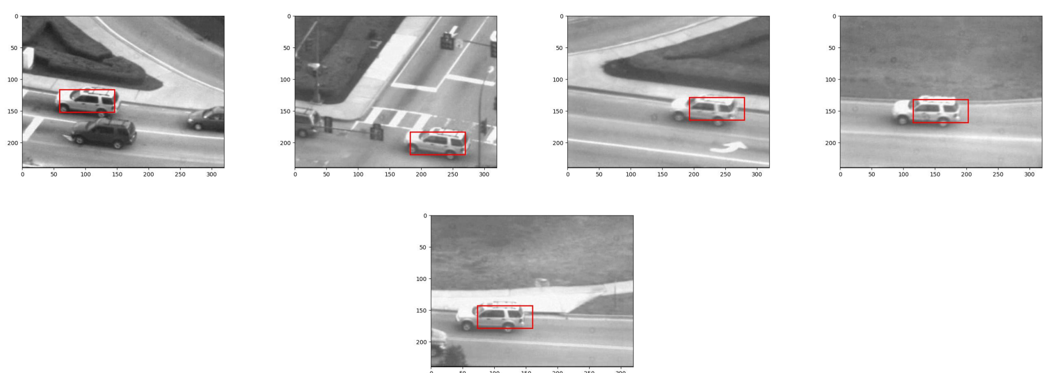
Landing



Car 1

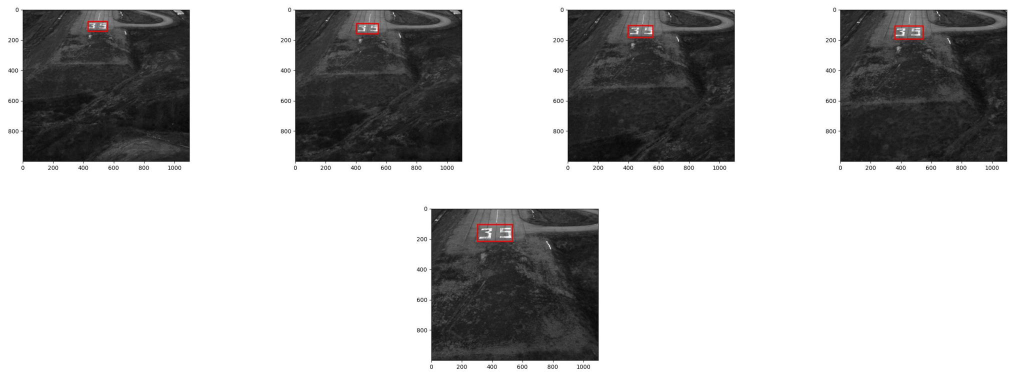


Car 2

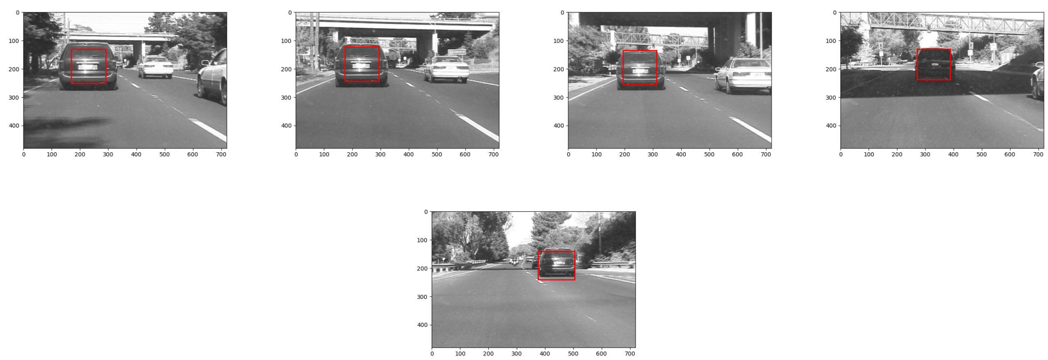


3.2 Lucas-Kanade Forward Additive Alignment with Affine Transformation

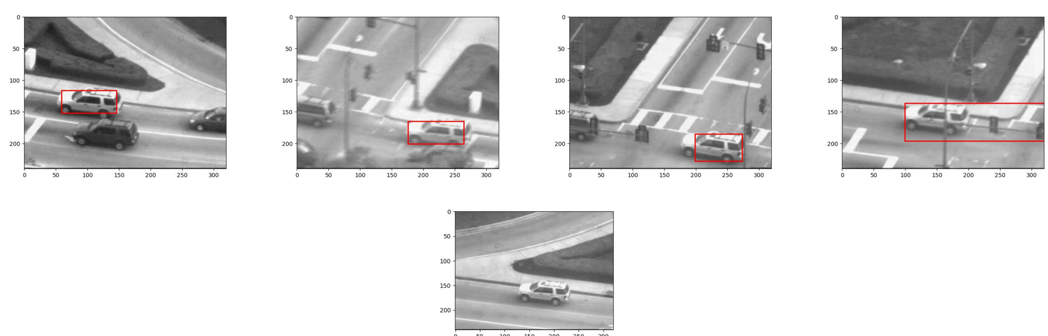
Landing



Car 1

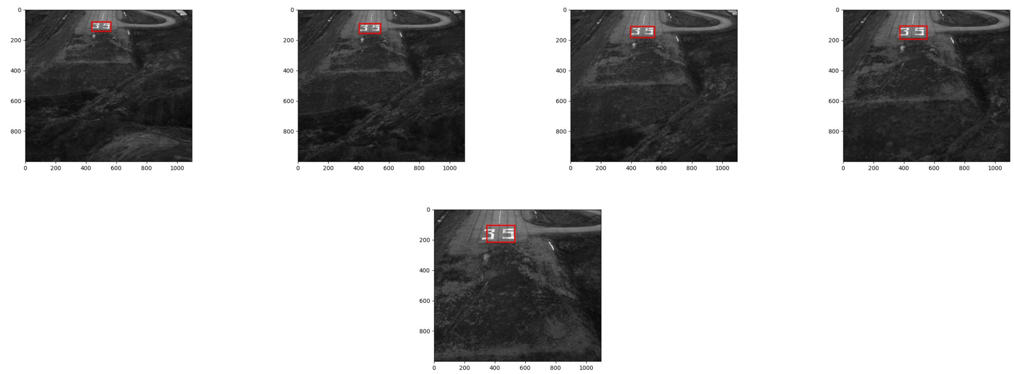


Car 2

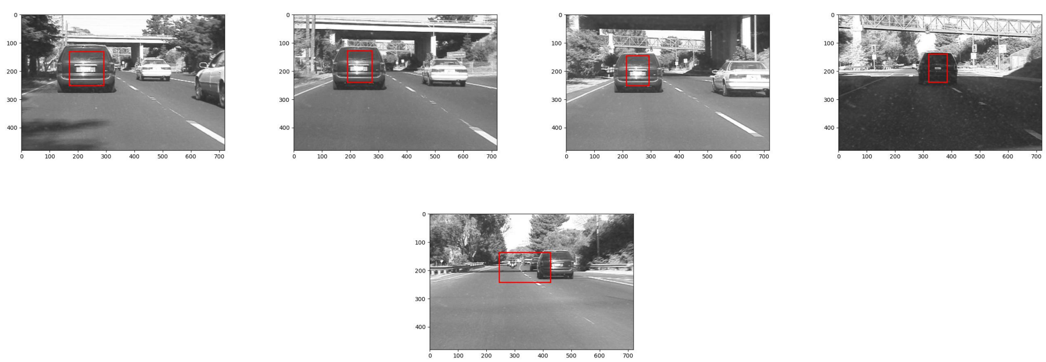


3.3 Inverse Compositional Alignment with Affine Transformation

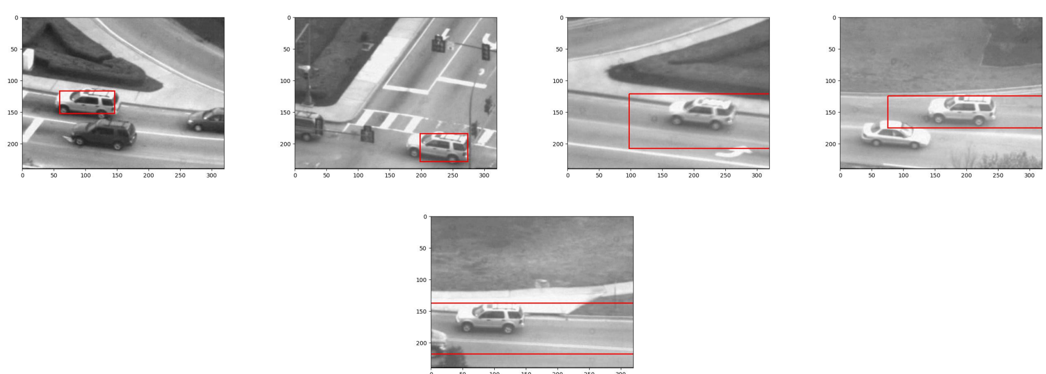
Landing



Car 1



Car 2



3.4 Testing the Algorithms

The Lucas-Kanade Forward method has the fastest performance and works well for small movements, but it's less accurate for larger areas or more complicated transformations like when the 35 on the landing strip get bigger as the camera gets closer. The way it's constructed seems better for simple translation not scaling or warping etc. It seems to handle the car1 video really well, keeping the tracking going even through changes with the shadows. It can also, because it doesn't handle scaling, handle the car2 video the best since once the car passes a big sign, it doesn't start taking that sign into account, it just moves over a small bit, So it technically doesn't handle it the best (its not robust) but it looks the best. These results happened because this method assumes that the translations between frames is small and intensities stay the same.

The Lucas-Kanade using Affine Transformation method is more flexible and accurate for things like translation and scaling, as it works well with keeping the 35 boxed in in the landing video and the same for car1, but it is slower by comparison to the normal LK method. However, it isn't robust to things like obstructions, so when a sign appears over the car in car2 video, the sign starts being taken into account in tracking and it follows that instead. At some point around the 200th frame the test script stops because the tracking has left the image space. Since the affine transformation allows it to handle more complicated transformations, it seems to overshoot when I new object enters the tracking space.

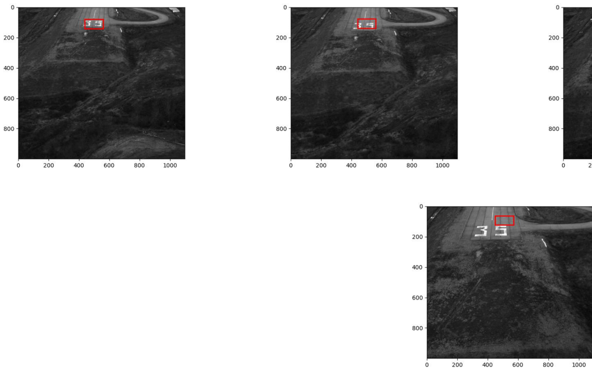
The Inverse Compositional with Affine method inversely warps the template to the image which seems to make it better computationally as it runs at a decent speed because it is pre-computing some of the values, though still slower than the first LK method. This also means it is more robust than the two previous methods. Though it still picks up on the sign in the car2 video, it doesn't replace the car with it, it adds it instead. With the landing and car1 videos, this method tracks them well but gets a bit off center when the shadows/contrasts change. I believe this method also stops a few frames short of the end of the video because the tracking space goes outside the image space.

4 Extra Credit

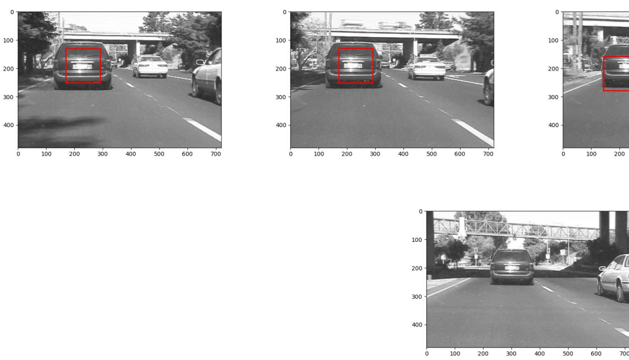
4.1 Adding Illumination Robustness

I did attempt to add illumination robustness but had some trouble. The results are on the following page.

Landing



Car 1



Car 2