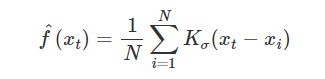
**Method Used** - Y=3

Video is a set of 2D Images. Think of each pixel of the image and observe it over time, we now have a sample of intensity values of that pixel over time. Let this sample be x1,x2,…,xN. We can obtain **the estimate of pixel intensity values by using kernel density estimation**. Given the observed xt at time t, we can estimate probability density function of the observation as,



where Kσ is a kernel function with bandwidth σ. In our case we will using Normal Kernel with



Here N is the sample size.

Now, the pixels which have the moving object will have higher pdf value evaluated in that frame compared to the parts which are still and hence assuming to form the background. We **choose the pixel values for a frame whose pdf exceeds some preset threshold as the background**.

* *Code Process ---*
* It reads all the images in the folder specified by imgs\_path . It sorts the images, reads them and stores them in a numpy array frames.
* The rows and cols variables store the number of rows and columns in each image
* The res array is initialized as a zeros array with the same number of rows and columns as the images
* The code then loops over the images in the FRAMES array (except the first image) to perform background subtraction using the kernel density estimation (KDE) technique
* The sig variable is calculated for each image based on its index k in the loop
* The DIV1 and DIV2 variables are calculated based on the SIG variable
* The code then loops over each pixel (i, j) in the image to calculate the KDE value for that pixel

i. The `r`, `g`, and `b` arrays store the red, green, and blue color values for all the previous images.

ii. The KDE value is calculated for the current pixel by summing up the exponentiated difference between the color values of the current pixel and the previous pixels, divided by `div2`.

iii. The calculated KDE value is then compared to the threshold `th`. If it is greater than or equal to `th`, the `res` array at the same position is set to 1.

* The resulting foreground image is then created by subtracting res from 1 and transposing the result. The image is then dilated using a square kernel of size KERNEL\_and iterations number of iterations
* The final image is then written to disk in the output folder with a filename "resk.png" where k is the index of the image
* All the foreground images are then processed to form a video named video.mp4
* To use constant weight, we need to assign a constant weight to each data point (previous color value of the pixel) in the calculation of the KDE value for the current pixel. This has been done by dividing the exponential term by the number of data points (previous color values) used in the calculation

**Failure Case:-**

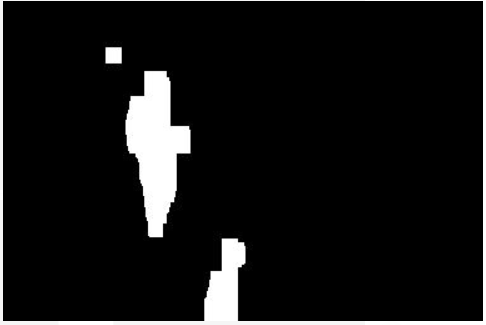
* Illumination Changes: Illumination changes can significantly affect the background subtraction results in KDE. The changes in brightness and contrast can lead to false positive detections and misidentifications of foreground objects
* Moving Background: Moving background such as swaying trees, moving water, etc. can result in the failure of KDE-based background subtraction
* Dynamic Background: Dynamic background, such as people or cars moving in the scene, can result in incorrect subtraction results
* Scene Cluttering: Scene cluttering, such as the presence of multiple objects in the foreground, can also result in false positive detections

**Comparison with GMM model :--**

* Accuracy: GMM can better model complex background distributions with multiple modes, while KDE can have difficulty modeling such complex distributions. However, KDE is computationally more efficient than GMM
* Smoothing: GMM often results in smoother background models than KDE, which can result in a better subtraction in noisy or cluttered scenes
* Adaptation: GMM can adapt to changes in the background over time, while KDE may not adapt as quickly



**KDE with constant weights**



**GMM with constant weights**

* Link for output videos **--** [OUTPUT VIDEOS](https://csciitd-my.sharepoint.com/:f:/g/personal/mcs222067_iitd_ac_in/Ejj39xIdpQVDojPzbM-_sNUBGSOQvNePWZuGW0liRNjMnw?e=RLeOJD)