EE 331 Probability and Random Processes

Programming Assignment

Ojas Mithbavkar, 17110083

Background Subtraction using Gaussian Mixture models

Dataset:

For this task, I downloaded a video of traffic from youtube. I cut the video length and reduced the resolution to 240p and fps to 10 to reduce the computational intensity. I read the video using opencv.

Algorithm:

The main essence of the algorithm is modelling the intensity of each pixel using mixture of normal distributions and updating the associated parameters based on the past intensities.

1. Initialize the means, variances and weights of every pixel. The sum of weights for every pixel should be 1.

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

where

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)}$$

Every channel RGB is assumed to be independent so that the Covariance matrix will be a diagonal matrix.

2. Given a current intensity of pixel X, we check if it ‘matches’ any of the K GMs. This matching ‘M’ is decided by Mahalonobis distance from the Kth GM. If a match is found, then the weights are updated as
where alpha is the learning rate.

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t})$$

The mean and variances are updated as follows for the matched ones:

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t$$
$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t)$$

where the second learning rate[3], ρ, is

$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k)$$

If there are no matches, we replace the least probable distribution with a new one with higher variance and low weight. A background pixel will have low variance as the object which it belongs to is relatively static in the background by definition.

3. To label a pixel as a background or a foreground one, we consider the following sum:

$$B = argmin_b \left(\sum_{k=1}^{b} \omega_k > T \right)$$

Where T is a hyperparameter threshold representing what portion of the data should be accounted for by the background. Given this label, we assign a pixel intensity to the foregrounds and background pixels to get an image wherein the background is subtracted from the original image.

Notes:

Better background subtraction can be achieved by hyperparameter tuning. The algorithm works realtime with every frame of the input.

References:

1. http://www.cse.psu.edu/~rtc12/CSE586Spring2010/papers/emBGsubtractAboutSandG.pdf
2. Stauffer, Chris, and W. Eric L. Grimson. "Adaptive background mixture models for real-time tracking." *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*. Vol. 2. IEEE, 1999.
3. https://www.video2edit.com/convert-to-mp4 used for fps and resolution manipulation
4. Test video https://www.youtube.com/watch?v=NcaGFp76BTY
5. https://github.com/artificertxj1/Background-Subtraction-with-Gaussian-Mixture
6. https://barkeywolf.consulting/posts/background-subtraction/