Assignment 1: Background Subtraction

Praveen Venkatesh
EE10B028

February 11, 2013

Abstract

This document supports a program that implements a background subtraction technique. The technique involves using a Gaussian Mixture distribution in order to model the noise and mode characteristics of a dynamic background. It is fairly robust to single-pixel noise and illumination changes in the background. The method strives to eliminate dynamic backgrounds such as moving branches, however this element is still in need of improvement.

1 The problem statement

A program is to be written, which effectively performs background subtraction on a given test video. A small clip for training the program is provided, which can be used to determine background pixel intensities and noise characteristics. The choice of algorithm is left open, however, it should be able to handle some of the following common issues related to background subtraction

- Illumination changes
- Gaussian and single-pixel noise
- Dynamic background effects

2 The algorithm

The algorithm that has been chosen in order to achieve the requirements is based on the Gaussian Mixture Model. This algorithm models the intensity variation in every pixel as a superposition of multiple gaussians. Each pixel will have its own set of gaussians, with different means and variances.

Given that the background fits a GMM, the training task reduces to one of fitting a given set of training vectors to a suitable set of gaussians. This is implemented using the method proposed by Stauffer et. al., as described below.

The task of actually subtracting the background from the given test video is also a simple one, wherein we estimate whether or not a given pixel lies in any of the gaussians in the corresponding background pixel. If it does, then it is a background pixel and if not, then it is taken to be a foreground pixel.

2.1 Training

The ideal method to fit data to a Gaussian Mixture is to use the expectation maximization algorithm. However, upon implementation, it was seen that this algorithm worked far too slowly in order to produce acceptable results. The EM algorithm tried to be more accurate at the expense of performing more computations.

A good trade-off turned out to be a method which updates the existing GMM parameters for each successive frame. This method promised to be much faster. Means, variances and weights are updated on an exponential basis for each successive frame. The exponentials used are parameterized by two important values: ρ, for means and variances and α, for weights. These values have been taken to be 0.001 and 0.1, but can be modified by editing the line in the program where they are defined.

The other parameter that has to be fixed is the number of gaussians to be used. This value has been taken to be 5, as it is unlikely that a pixel will see more than 5 different modes in terms of illumination variation and dynamic background effects.

2.2 Subtraction

The key parameter that is to be chosen for subtraction is the threshold or tolerance, v. This value determines the number of standard deviations from the mean within which a pixel will be considered to be a background pixel. This parameter was empirically tuned to a value of 7.

2.3 Miscellaneous parameters

Other tweaking measures include the ability to skip frames, both while training and while testing. Skipping frames while training gives a more diverse set of training vectors, and appears to produce better results on average. However, dynamic background effects are always present. Avoiding skipping frames while testing produces a better best-case result but a worse worst-case result. The effect of the dynamic background persists, and is eliminated completely in some frames, but not at all in others.

3 Results

The current implementation manages to handle noise and illumination changes quite well. This is achieved by:

- Using the GMM to model the noise distribution as a gaussian, while permitting multiple intensities through superposition
- Choosing an appropriate threshold value that maximally eliminates noise, while maintaining the integrity of foreground objects

However, there is room to improve in the manner in which dynamic backgrounds are dealt with. Some proposals for improvement include:

- Using some kind of smoothing function on the final image. This would work well if frame-skipping was used while training, so that the noise due to dynamic background effects would be scattered on average. That would then enable median smoothing, for example, to remove stray pixels and blobs, while consolidating foreground objects.
- Increasing the speed of the algorithm (by multi-threading, for instance) so as to enable the use of a larger number of training frames within a reasonable time interval.