Computer Vision I

Project 1
September 20, 2019

Due on October 4, 2019

Motion Detection Using Simple Image Filtering

The purpose of this project is to get familiar with manipulating images and image filtering techniques.

1. **Motion Detection**

   In this project you will explore a simple technique for motion detection in image sequences captured with a stationary camera where most of the pixels belong to a stationary background and relatively small moving objects pass in front of the camera. In this case, the intensity values observed at a pixel over time is a constant or slowly varying signal, *except* when a moving object begins to pass through that pixel, in which case the intensity of the background is replaced by the intensity of the foreground object. Thus, we can detect a moving object by looking at **large** gradients in the *temporal* evolution of the pixel values.

2. Project Requirements:

   (a) Write a program to implement the above ideas. Sample videos to test your program are available in blackboard. The basic outline of your program should be as follows:

       i. Read in a sequence of image frames and make them grayscale.

       ii. As enough frames are available, apply a 1-D differential operator at each pixel to compute a *temporal* derivative.

       iii. Threshold the absolute values of the derivatives to create a 0 and 1 mask of the moving objects.

       iv. Combine the mask with the original frame to display the results.

   (b) You should explore the following variations on the above basic outline:

       i. For the temporal derivative filter try a simple $0.5[-1, 0, 1]$ filter and a 1D derivative of a Gaussian with a user defined standard deviation tsigma. Compare results running these filters; for the Gaussian filter try increasing values of tsigma. Discuss your results.

       ii. Try applying a 2D spatial smoothing filter to the frames before applying the temporal derivative filter. For the spatial smoothing filter try 3×3, 5×5 box filters and 2D Gaussian filters with a user defined standard deviation ssigma. Run the program using different amounts of smoothing. Discuss your results. Are there values that make it work better? Explain what you mean by better.

       iii. Vary the threshold to get the mask and see what works best. Design a strategy to select a good threshold for each image. Hint: Most pixels belonging to the background change little and thus their temporal gradients are close to zero. You can model these values as Gaussian zero mean noise and estimate the standard deviation of this noise.

   (c) Write a report. The report should include:

       i. Abstract, description of algorithms, experiments, values of parameters used, observations and conclusions.

       ii. Representative images that you tested the algorithms with as well as representative outputs showing both "good" and "bad" results.

       iii. An appendix with your code