A Benchmark Study on Algorithms for Motion Detection

Si Chen
Department of Computer Science
The George Washington University
sichen@gwu.edu

Bjoern Cheng-Yi
Department of Computer Science
The George Washington University
yicheng0922@ gwu.edu

Abstract

In this paper, we compare different approaches to detecting motion in videos using optical flow images as input. Algorithms analyzed in this comparison include background subtraction and other segmentation algorithms such as k-means and mean shift. This study aims to highlight background subtraction as the algorithm with the highest accuracy and presents future work to be considered in this topic.

1 Introduction

Currently, there is a high concentration of research power focused on the application of computer vision algorithms to self-driving automobiles. For example, Google has been involved in self-driving cars since 2009 and other companies such as Uber and Baidu are following in its footsteps. Since the technology behind autonomous vehicles is relatively new, researchers face many challenges in approaching the data output by the various cameras mounted on these vehicles. One such problem is the issue of detecting objects coming towards the vehicle, an application that would allow cars to slow down or turn away from such objects to avoid accidents.

This benchmark will focus on a key part of that challenge by evaluating algorithms' ability to detect motion by segmenting optical flow images into clusters representing moving objects and the background. In this project, we extract the optical flow data from a video and then, segment the data to find the moving object using various algorithms. Each of these algorithm's output accuracies will be analyzed with respect to the ground truth annotations.

1.1 Optical Flow

In an image, optical flow is the pattern of motion of an object caused by relative motion. We decided to test optical flow versions of the input video rather than the original RGB frames because the optical flow images are an indication of the moving objects in the frame. Given that our dataset uses a fixed camera, the unmoving background should return pixel values of 0, indicating no movement, whereas the moving foreground will be highlighted in other colors.

Figure 1: Example Output of Optical Flow

Furthermore, current approaches towards detecting motion in driverless vehicles use expensive sensing systems, which cost anywhere from $5,000 for a simple depth imaging camera to over $40,000 for a single LiDAR sensor. Improving optical flow and creating a method of calibrating information output by optical flow with motion data would allow for an inexpensive method of approximating motion from RGB images. The following algorithms are tested on their accuracy in extracting motion from optical flow images to support this goal.

2 Algorithms

In this benchmark, we will compare different methods for tracking moving objects on optical flow data. The RGB optical flow frames will be used to compare a variety of algorithms that range from background subtraction to clustering-based segmentation.

2.1 Background Subtraction

Background subtraction extracts an image's foreground, in this case, the moving object, by compare current frames of a video with a reference background model. The model of the background is created using initial frames of the background that include noise and slight shifts in a background. This model is then compared with new frames of the scene and subtraction occurs to show the difference in the output foreground mask. The prevalence of background subtraction in applications to video surveillance indicates its effectiveness as a method for identifying moving objects. However, this method has no understanding about the pixels labeled foreground—there is no grouping of these foreground pixels into object blobs and no post-processing to determine whether or not residual noise has been included in the foreground mask.

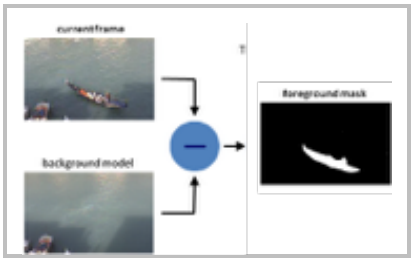

Figure 2: Example Output from Background Subtraction

2.2 K-Means Segmentation

Since RGB optical flow will highlight areas of the frame that were moving at the same rate in the same color, k-means clustering based segmentation will allow us to segment the image based on color. Analysis of this algorithm will require us to test various k values and cluster initializations. These motion videos contain different moving objects and we are interested in seeing how the clusters will be able to update their location with respect to new objects in a scene. K-means clustering will cluster an image by color into different segments, but its adaptability is hindered by the need to choose a set number of clusters.

Figure 3: Example Output from K-Means Segmentation

2.3 Mean Shift Segmentation

We will also test mean shift as a basis for segmentation. This algorithm approaches the problem of segmentation by homogenizing local groupings, replacing each pixel with the mean of the pixels in a diameter range that has a RGB value within a certain range. In this sense, it is very similar to k-means segmentation. The main differences lie in the fact that cluster locations do not need to be initialized and neither do the number of clusters, although they can be, by iterating over an image with mean shift until only a certain number of clusters remain. Mean shift clusters

Figure 4: Example Output from Mean Shift

2.4 Watershed Algorithm

The watershed algorithm is another common image segmentation algorithm that is commonly used for biomedical applications in identifying individual cells based on coloring. Watershed focuses on being able to separate overlapping segments into individual segments. It does so by identifying "ridges" in an image and separating basins created by the ridges into separate segments.

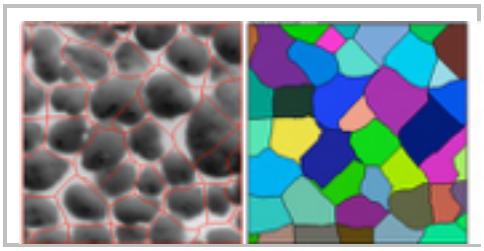

Figure 5: Example Output from the Watershed Algorithm

2.5 Graph-Based Segmentation[1]

Our last approach to segmentation is graph-based. This algorithm represents the segmentation problem in terms of a graph where each node in the set of vertices corresponds to a pixel in the image and each edge connects pairs of neighboring pixels. Based on the property (ex. HSV value) of the pixels, the edges have varying weights, and pixels are classified based on the weighted edges into segments. This is a more recent approach to segmentation that attempts to capture perceptually important groupings in images that do not have to be spatially local, using edge weights to group in a greedy method. In other words, pixels do not have to be near one another in an image to be grouped in the same segmentation. Furthermore, this algorithm runs in O(nlogn) time, which allows for this algorithm to be run as fast as the rate of modern video frames per second.

Figure 6: Example Output from Graph-Based Segmentation

4 Dataset

The dataset that will be used in this project will be the video dataset from the Background Models Challenge (BMC)[2]. This dataset contains videos with manual annotations on given frames. It also provides over 120 initial frames of just the data, which will allow us to compare the background subtraction algorithm, which needs over 115 frames to build a model of the background, with the segmentation algorithms.

For the clustering algorithms that require a number of clusters to be pre-chosen, we will evaluate them against one another with different numbers of clusters and different initializations of the cluster centers. From the best cluster number choice and initialization for the clustering algorithms, we will compare their accuracies and speeds to the other algorithms' accuracies and speeds. We will also evaluate the performance of the various algorithms on videos from high quality, which will be the videos from the dataset, to low quality, which will be the videos from the dataset that are compressed.

The BMC dataset we are using contains ground truth annotations that will allow us to calculate the accuracies of the motion clusters. We chose this dataset over other benchmark tracking datasets due to the detailed labels given. In most tracking datasets, only bounding boxes are given for objects rather than a pixel-by-pixel label of "moving" and "nonmoving" objects in a scene. We will use f-scores to evaluate the precision and recall of each algorithm in being able to identify the moving objects in the frame.

5 Results

Qualitative

The Mean Shift algorithm performance is good. The result image has a much smoother color change, and much less noises appeared on the blobs. The main problem is that the algorithm took a very long time to process the images, which makes it hard to be used to avoid accidents.

The Watershed algorithm performance is bad. The result images turn out to be meaning less segmentation. Watershed looks for an area that is darker and classifies it as an edge, but the optical flow images do not have such edges, causing the algorithm to generate meaningless result.

Quantitative

To analyze the data quantitatively, we used F-Score Comparison and the following is the result for that. Watershed is not analyzed because the results are meaningless, and Graph Based is not analyzed due to the fact that the color of the segmentation keeps changing and it is impossible to analyze it with F-Score. The result of the comparison is shown in Table 1.

Table 1: F-Score Comparison

Algorithm	**F-Score**	
Background Subtraction	0.674	
K-Means	**K Value**	**F-Score**
	2	0.437
	3	0.485
	4	0.499
	5	0.510
Mean Shift	0.465	
Watershed	N/A	
Graph Based	N/A	

The importance of this work is twofold: 1) it establishes the Background Subtraction algorithm as the best method of detecting motion and 2) it highlights some of the current issues seen in optical flow. Given the relevance of this work to driverless vehicles and its cost-effective method of displaying motion information, it is important for us to further expand on the use of optical flow as the input for motion segmentation.

References

[1] Felzenszwalb, P.F. & Huttenlocher, D.P. (2004) Graph-based image segmentation. *International Journal of Computer Vision* 59 (2), 167 - 181.

[2] Vacavant, A., Chateau, T., Wilhelm, A., & Laurent, L. (2012) A Benchmark Dataset for Foreground/Background Extraction. *ACCV 2012, Workshop: Background Models Challenge,* LNCS 7728, 291-300.