Distributed Image Segmentation

Andy Li, Yulun Tian, EE 122 Spring 2016

Introduction

In this research project we focus on image segmentation, an important topic in today's field of image processing and also a cornerstone for more advanced technologies such as computer vision. While there are mature algorithms for solving this problem already, we seek to optimize their performance by deploying them in a distributed setting. During the process we evaluate the accuracy of our distributed program, and we also study its structure and speed up pattern to determine where efforts should be put for further optimization.

Algorithms

Image segmentation is the process in which we group similar pixels in an image into clusters. In this project, the main algorithm we consider is the mean shift segmentation algorithm[1][2]. As a reference algorithm, we also consider the efficient graph-based segmentation[3].

Mean Shift Segmentation

The algorithm contains a shifting stage and a clustering stage. In the shifting stage, each pixel in the image is represented as a point in the higher dimension feature space (x, y, r, g, b), where x, y describe the location and r, g, b describe the color. Furthermore, each point is viewed as a sample from an underlying probability density function. For each point, our goal is to find its mode (i.e. the nearest local maximum in the pdf).

To do this, we shift the point along the direction of the gradient, which is estimated as $\nabla f(x) \propto (\frac{1}{|S_{x,hs,hr}|}\sum_{x_i \in S_{x,hs,hr}} x_i) - x$. Here x is a point in the feature space, and $S_{x,hs,hr}$ is the set of neighbouring points that is within hs in distance and within hr in color. It can be proved that after repeated shifting, the point will converge to its mode [2].

In the clustering stage, we simply cluster similar modes into a group. Then, each group is outputted as a segment in the final result.

Efficient Graph-based Segmentation

Given an image, construct a graph G = (V, E), where V is the set of pixels in the image. For each pair of neighbouring pixels, add an undirected edge to E, whose weight is the color difference.

In G, we represent a cluster by the MST that connects all pixels in the cluster. Initially each pixel forms its own cluster.

Iterate through the edges in increasing order. For edge e, consider the two clusters C_1 and C_2 joined by e. Merge these two clusters if e is smaller than $\min(l_1 + \frac{1}{|C_1|}, l_2 + \frac{1}{|C_2|})$, where l_1 and l_2 are the largest edge in C_1 and C_2's MST respectively. Return all remaining clusters after the iteration completes.

Distributed Settings

In this part we focus on implementing a distributed version of mean shift segmentation. To do this, we turned to the PaaS (Platform as a Service) Amazon Web Services, AWS[4]. We used the following AWS Services:

- Amazon EC2 (Elastic Compute Cloud) - Virtual Servers in Cloud
- Amazon S3 (Simple Storage Service) - Scalable Storage in Cloud
- Amazon SQS (Simple Queue Service) - Message Queue Service

We designed and wrote code for distributed algorithm calculations, communication between the main machine and other nodes, and communication to an external subscriber. First we describe the external communicating architecture. The front-end dashboard allows submissions of images and parameters to S3, the common file store and subscribes to SQS which feeds it messages on the updates of progress. The EC2 Machines publish their progress to SQS as the algorithm is running. This reflects a paradigm of communication architecture known as pub-sub (publisher/subscriber).

Now we would like to describe the inter-machine communication structure. When an image is observed in our common file store (S3), the machines begin to run the mean-shift stage of our algorithm. The output is then fed back to the common file store, and the main machine then takes in the information for the serial processing segment of the algorithm. The main machine then runs the serial clustering of the algorithm, and feeds the output back to our common file store.

Our architecture exhibits fundamental ideas of communication. The common file store, represents broadcasting of messages, the messaging of progress exhibits the publisher-subscriber paradigm, and the distribution of machines ultimately exhibits how based on a set of agreed protocols and rules, efficient and effective communication can occur. As a result, we were able to build a proof of concept of SaaS (Software as a Service) upon a PaaS.

Analysis

By comparing the output matrices of one-machine vs. distributed running of our algorithm, we have indeed confirmed that the two methods do produce identical results. Moreover, we have confirmed that in general, mean shift segmentation yields better result than graph-based segmentation. By running experiments, we have seen that as expected, adding more machines to run the algorithm greatly speeds up the processing of the mean-shift stage. There was no significant difference observed when running the clustering phase of the algorithm. Furthermore, running on different thresholds of color hr, or distance hs, was not observed to significantly alter the running time. See presentation for more analysis.

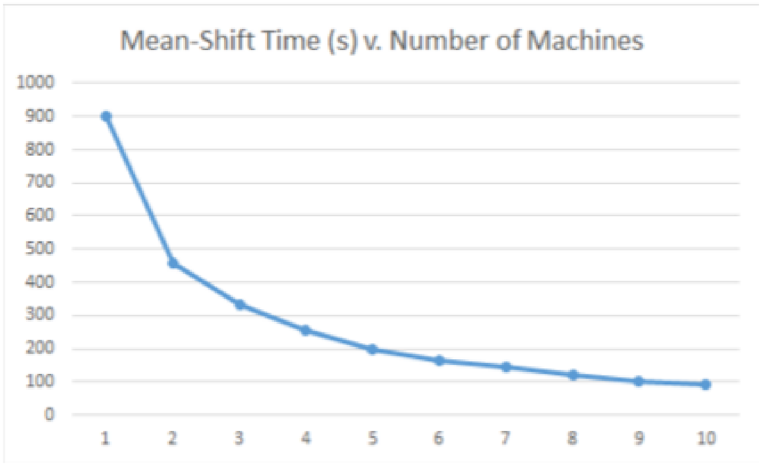

Figure 1: Distributed mean shift segmentation speed up pattern

References

[1] C. Pantofaru, M. Hebert, "A Comparison of Image Segmentation Algorithms", The Robotics Institute Carnegie Mellon University.
[2] D. Comaniciu, P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis", IEEE Trans. on Pattern Analysis and Machine Intelligence.
[3] P. Felzenszwalb, D. Huttenlocher, "Efficient Graph-Based Image Segmentation", Intl Journal of Computer Vision.
[4] Amazon Web Services, `http://aws.amazon.com/`