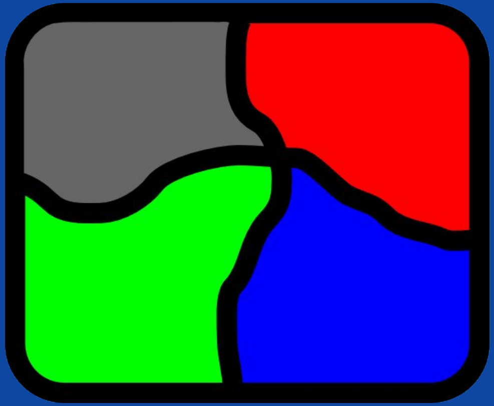
K-Image Segmentation With Modified Network Flow



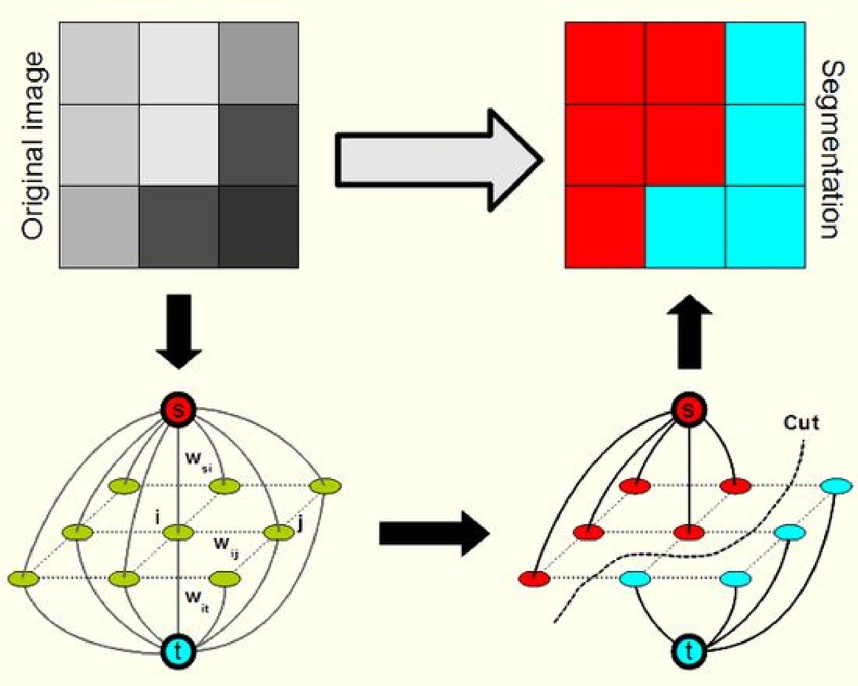
Abhyudaya Alva Furkan Eris Aditya Narayan Hien Duc Nguyen

EC504 Term Project

BOSTON UNIVERSITY

Image Segmentation Initial Approach

- Edges from source to all pixels with capacity “foreground likelihood”
- Edges from pixels to sink with capacity “background likelihood”
- Fixed Penalty between all neighboring pixels
- MinCut using Ford Fulkerson



Ford Fulkerson Implementation

- Represent Graph as Adjacency List instead of Adjacency Matrix.
- Adjacency List using `std::unordered_map<int,std::unordered_map<int,int> >` for O(1) access.
- Edmond Karp’s Algorithm with Iterative Breadth First Search to find Augmenting Paths
- Overall Complexity $O(VE^2)$

K-means Clustering

- NP-Hard Problem
- Given a set of observations ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k ($\leq n$) sets $\mathbf{S} = \{S_1, S_2, \ldots, S_k\}$ so as to minimize the within-cluster sum of squares. In other words, its objective is to find:

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2$$

where $\boldsymbol{\mu}_i$ is the mean of points in S_i.

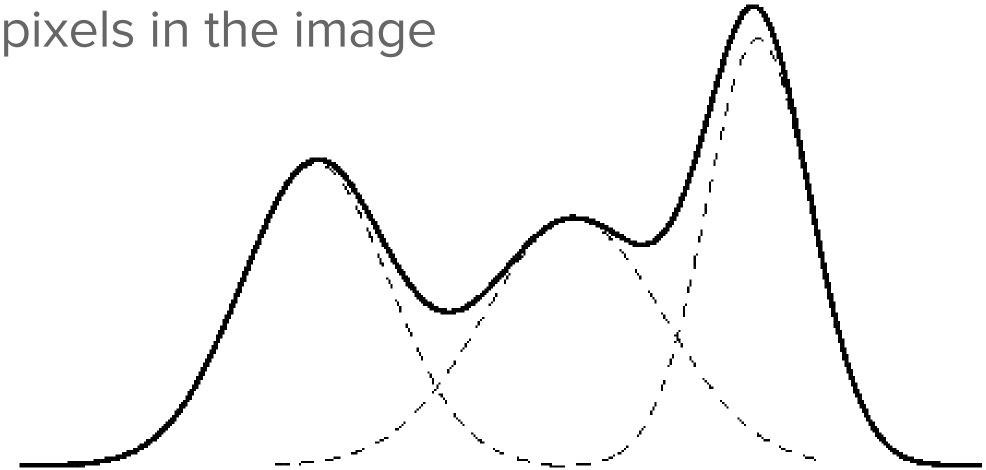
K-means Algorithm

Given an initial set of k pixels, the algorithm proceeds by alternating between two steps:

- Assignment step: Assign each pixel to the cluster whose mean yields the least within-cluster sum of squares
- Update step: Calculate the new means to be the centroids of the pixels in the new clusters

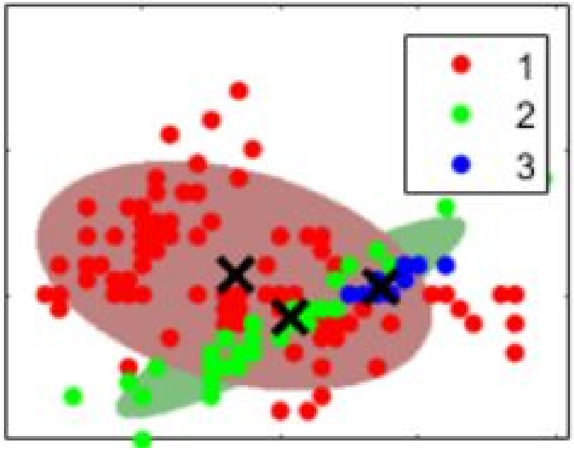
Clustering Improvements GMM

- Gaussian Mixture Models: Soft Labels: Each Iteration Assignment of Probability Distribution
- A model corresponds to the mixture distribution that represents the probability distribution of pixels in the image



Clustering Comparison

K-means



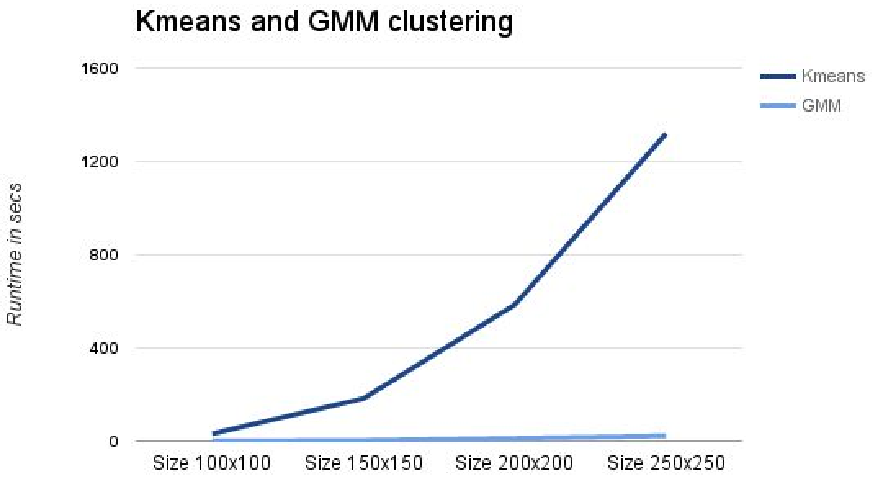
Gaussian Mixture Models

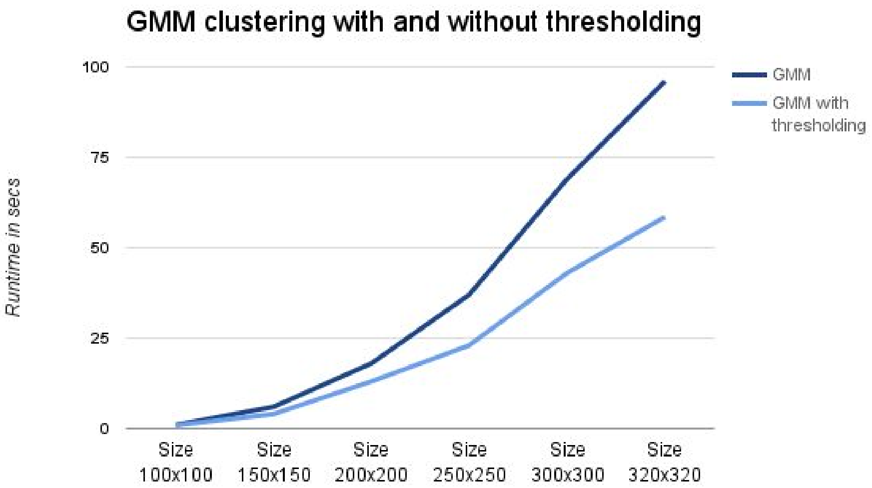
Algorithm Improvements Timeline

1. Clustering using K-means + Network Flow using Ford Fulkerson + penalty computation to all neighbours - (35x35 image took ~10 mins)
2. Ford Fulkerson implementation using maps and vectors - (128x128 image took ~ 5mins)
3. Penalty Computation using thresholds - (128x128 image took ~3mins)
4. Ford Fulkerson implementation using hash maps and Clustering using Gaussian Mixture Models - (500x500 image takes ~ 3mins)

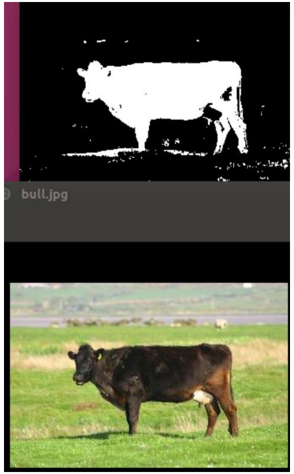
BOSTON UNIVERSITY

Runtime Improvement Techniques





Results for Two Segments

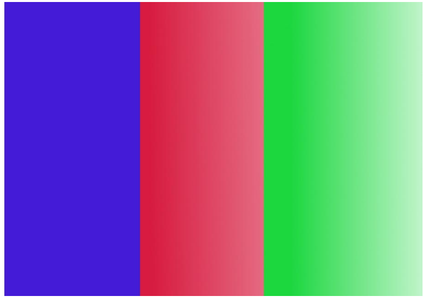


Using K-means Clustering

Using GMM Clustering

Penalty Thresholding

Extension to K-Segments

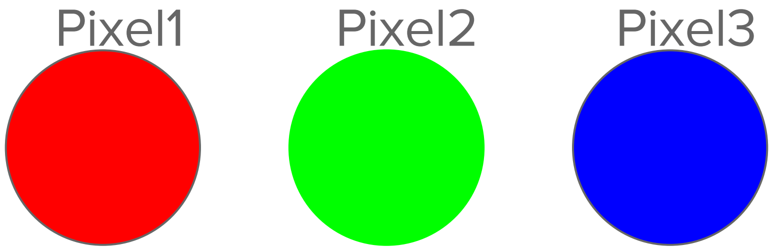


Our novel approach to k-segments

Method of OVA

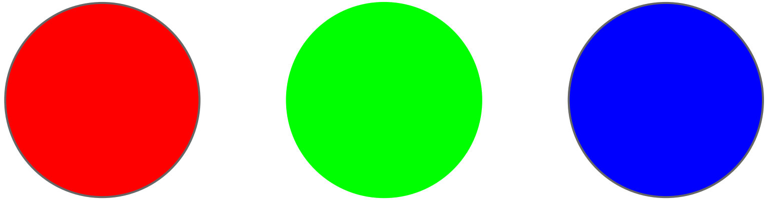
- Continuity
- Linear Time Increase Makes Complexity of Code Naively $O(k*|V|*|E|^2)$

k=3



RGB Values Pixel1: (256,0,0) Pixel2: (0,256,0) Pixel3: (0,0,256)

K-Segment Example



First Step: Get Probability Values For Each Pixel From GMM

Pixel1: (10, 0, 0) Pixel2: (0, 10, 0)   Pixel3: (0,0,10)

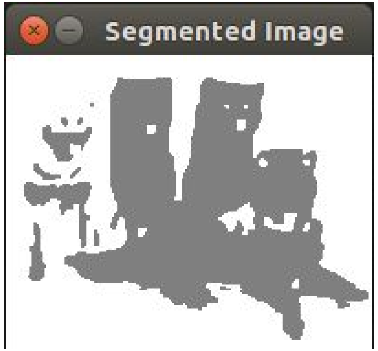
Step 2: Ford Fulkerson for Segment1 vs all other Segments

Pixel1:(10,0) Pixel2(0,10) Pixel3: (0,10)

FF Returns Labels for Pixel1: Foreground1 Pixel2: Background Pixel3: Background

Step3: Repeat Step2 Removing Pixels in the Foreground for k-1

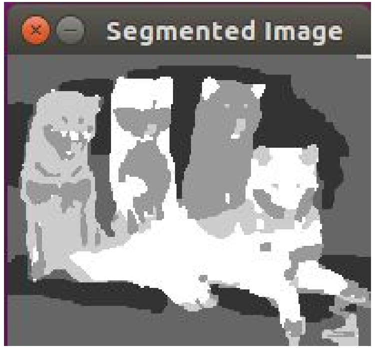
Results for K-Segments



k=2

k=4

k=3



k=5

Acknowledgment

We would like to thank Prof. Kulis for this valuable class.

References

[1] BayesianGaussianMixture. “2.1. Gaussian Mixture Models — Scikit-Learn 0.18.1 Documentation.” 2010. Accessed December 12, 2016. http://scikit-learn.org/stable/modules/mixture.html.

[2] Greig, Dorothy M., Bruce T. Porteous, and Allan H. Seheult. "Exact maximum a posteriori estimation for binary images." Journal of the Royal Statistical Society. Series B (Methodological) (1989): 271-279.