Adaptive Superpixel Cuts for Hyperspectral Images

Aleksandar Popovic

February 2024

Abstract

Blind segmentation in hyperspectral images is a challenging problem. Many traditional methods suffer from poor identification of materials and expensive computational costs, which can be partially eased by trading the accuracy with efficiency.

In this paper, we propose a novel graph-based algorithm for segmentation in hyperspectral images. Utilizing the fact that pixels in a local region are likely to have similar spectral features, a pre-clustering algorithm is used to extract the homogenous regions, called superpixels. After extracting the superpixels, a weighted graph is constructed with the weights representing both the spectral similarity and spatial distance between each superpixel and its neighbors. The normalized graph cuts algorithm is then used to perform an initial segmentation of the image. To effectively extract the material information in the superpixels, the mean spectra in each segment is used to estimate the abundance of each endmember in each superpixel using a graph regularized hyperspectral unmixing algorithm. The resulting abundance information is used as a supportive feature, which when combined with the spectral features, form a new spectral feature vector for each superpixel. Using this new feature vector, the weighted graph is once again constructed and the normalized cuts algorithm is applied, resulting in a final segmentation of the image.

Experiments on a real hyperspectral datasets illustrate great potential of the proposed method in terms of accuracy and efficiency.

Contents

1 Introduction **3**

1.1 Motivations . . . 3

2 Background **4**

2.1 Hyperspectral Imaging: Basics . . . 4

2.1.1 The Hyperspectral Cube . . . 4

2.2 Superpixel Generation . . . 5

2.2.1 Simple Linear Iterative Clustering . . . 5

2.3 Spectral Clustering . . . 7

2.3.1 The Laplacian Matrix . . . 7

2.3.2 Normalized Cuts . . . 7

2.4 Hyperspectral Unmixing . . . 8

2.4.1 Alternating Direction Method of Multipliers . . . 8

2.4.2 Global Variable Consensus Optimization using ADMM . . . 8

2.4.3 Abundance Estimation . . . 8

2.4.4 Solution Constraints and Regularization Terms . . . 8

3 Adaptive Superpixel Cuts **9**

3.1 Dataset Preprocessing . . . 9

3.1.1 Singular Value Decomposition . . . 9

3.1.2 Layer Normalization . . . 9

3.2 Algorithm Overview . . . 10

3.2.1 Superpixel Generation . . . 10

3.2.2 Construction of the Affinity Matrix $\mathbf{W}$ . . . 10

3.2.3 Normalized Cuts Algorithm . . . 10

3.2.4 Graph Regularized Fully Constrained Unmixing . . . 10

3.2.5 Feature Vector Creation . . . 10

3.2.6 Parameter Selection . . . 10

4 Experimental Results **11**

4.1 Implementation Details . . . 11

4.2 Geospatial Testing Datasets . . . 12

4.2.1 Samson Datasets . . . 12

4.2.2 Salinas Datasets . . . 12

4.3 Algorithm Evaluation . . . 13

4.3.1 Qualitative Evaluation on Samson . . . 13

4.3.2 Quantitative Evaluation on Salinas . . . 13

4.4 Algorithm Comparison . . . 14

5 Applications in Biomedical Imaging **15**

5.1 Dr Yucel Motivations . . . 15

5.2 Results on Biomedical Hyperspectral Images . . . 15

6 Conclusions **16**

1 Introduction

1.1 Motivations

2 Background

2.1 Hyperspectral Imaging: Basics

In hyperspectral imaging, near contiguous narrow band spectral information is measured for each spatial pixel of an image collected over a scene. Spectral information can be quantified by multiple measures. Traditionally, spectral radiance, being the energy emitted or reflected by a surface over a large number of spectral wavelengths, has been the measure of choice for physical applications. Utilizing this spectral information, physical and chemical properies of materials can be deduced and insights can be made about the overall composition of the images, resulting in applications across various fields.

Analysis in hyperspectral images has traditionally been a computationally expensive and difficult task due to algorithms scaling in both the spatial and spectral resolution of the images. This section will focus on building a relevant background for common preclustering, abundance estimation, and segmentation techniques that are common in hyperspectral image analysis.

2.1.1 The Hyperspectral Cube

In traditional, RGB based imaging systems, an image can be represented by a 3-dimensional tensor of shape $(n_x, n_y, 3)$, where the last dimension corresponds to the color channel the image was captured in. From a mathematical point of view, a hyperspectral image, denoted by $\mathbf{X}$, is a tensor of shape (n_x, n_y, n_λ) with nonnegative entries. Each pixel in the tensor is represented using a vector $\mathbf{x} \in \mathbb{R}_+^{n_\lambda}$. From a physical point of view, the first two dimensions in $\mathbf{X}$ represent the spatial coordinates of the pixels, while the last dimension represents the specific wavelength band the spectral intensity, reflectance or transmittance measurements were taken at.

The tensor $\mathbf{X}$ can be transformed into a two-dimensional matrix of shape (n_p, n_λ), denoted $\mathbf{X}_f$, where $n_p = n_x n_y$ is the total number of pixels in the image, with each column representing the pixel at index (i, j) in $\mathbf{X}$, arranged across the first spectral dimension, then the second. Formally, $\mathbf{X}_f = \left[\mathbf{x}_{(0,0)}, \cdots, \mathbf{x}_{(n_x,0)}, \cdots, \mathbf{x}_{(0,n_y)}, \cdots, \mathbf{x}_{(n_x,n_y)}\right]$.

2.2 Superpixel Generation

As mentioned in Section (2.1), high spectral resolution is a common constraint in hyperspectral image analysis, with time complexity of the algorithms commonly scaling polynomially with the number of pixels in the image.

Before the introduction of neural network based solutions in computer vision applications, there was interest in preclustering images into locally homogeneous regions called superpixels [superpixel ref]. Addressing the main motivation of reducing the overall granularity of the data, superpixels are also shown to preserve spectral information, adhere to spatial features in the image, and introduce robustness against noise in subsequent analysis tasks.

In non-hyperspectral applications, the superpixel algorithm uses spectral information in the 3-dimensional CIELAB colorspace and spatial information in the form of pixel coordinates.

2.2.1 Simple Linear Iterative Clustering

In this section, we will introduce the Simple Linear Iterative Clustering (SLIC) algorithm. The algorithm is a special case of the k-means algorithm adapted to the task generating superpixels in a 5-dimensional space, where the first 3 dimensions correspond to the the pixel color vector in the CIELAB colorpsace, and last 2 dimensions correspond to the spatial coordinates (i, j) of the pixel in the image.. Formally, we restructure each pixel $\mathbf{x}_{(i,j)} = [\mathbf{x}_l, \mathbf{x}_a, \mathbf{x}_b]$ into the form $\tilde{\mathbf{x}}_{(i,j)} = [\mathbf{x}_l, \mathbf{x}_a, \mathbf{x}_b, i, j]$. With this modified feature vector $\tilde{\mathbf{x}}$, we incorporate both spectral and spatial information into the clustering, however, while the spectral information has bounds on it's values, the spatial information depends on the size of the image.

Taking as an input the desired number of superpixels n_s, for an image with $n_p = n_x n_y$ pixels, each superpixel would be composed of approximately n_s/n_p pixels. Assuming the superpixels lie on a grid, a superpixel centroid would occur at every grid interval $S = \sqrt{n_s/n_p}$. At the onset of the algorithm, a grid of n_s superpixel centers $\mathbf{C}_n = [\mathbf{c}_l, \mathbf{c}_a, \mathbf{c}_b, i, j]$ where $n = 1, \cdots, n_s$ are sampled across the image with regular grid intervals S. To avoid sampling noisy pixels, clusters are moved to the lowest gradient position in a 3×3 neighborhood where the image gradient is calcuated, using the original spectral vector x in the CIELAB color space as:

$$\mathbb{G}(i, j) = \|\mathbf{x}_{(i+1,j)} - \mathbf{x}_{(i-1,j)}\|^2 + \|\mathbf{x}_{(i,j+1)} - \mathbf{x}_{(i,j-1)}\|^2 \tag{1}$$

After initialization, a modified distance measure is proposed to enforce color similarity and spatial extent within the superpixels. Since the approximate area of each superpixel is S^2, it is assumed that pixels associated with a superpixel lie within a $2S \times 2S$ neighborhood of the superpixel centroid. Introducing the parameter m to control the compactness and shape of the superpixels, the modified distance is then calculated as

$$\mathbb{D}(x, y) = \|\mathbf{x}_{lab} - \mathbf{y}_{lab}\|^2 + \frac{m}{S}\|\mathbf{x}_{ij} - \mathbf{y}_{ij}\|^2 \tag{2}$$

Each pixel is the image is associated with the nearest cluster whose search area overlaps this pixel. After all pixels are associated with a cluster, a new center is computed as the average feature vector of all the pixels belonging to the cluster. This is repeated for a set number of iterations. After exhausting all iterations, a final step is performed by relabelling disjoint segments with the labels of the largest neighboring cluster. This step is optional as disjoint segments tend to not occur for larger inputs of n_s and m.

Algorithm 1: SLIC Superpixel Algorithm

```
Input: $\mathbf{X}_f$, $m > 0$, $n_s > 0$, $n_{\text{iters}} > 0$
Initialize: $\mathbf{C}_n = [\mathbf{c}_l, \mathbf{c}_a, \mathbf{c}_b, i, j]$ where $n = 1, \cdots, n_s$ by sampling pixels at regular
  grid intervals $S$. Perturb cluster centers to lowest gradient position in a $3 \times 3$
  neighborhood according to (1)
for $k = 1$ to $n_{iters}$ do
    Assign best matching pixels from a $2S \times 2S$ neighborhood around clusters $C_k$
      according to distance measure (2). Compute new cluster centers according to
      average of all pixels belonging to cluster.
end
Optional: Relabel disjoint segments.
```

The SLIC algorithm is shown to produce overall meaningful and noise-robust segments in traditional computer vision applications. This algorithm proves useful in Section 3.2.1 when adapted as a pre-clustering step in the hyperspectral domain.

2.3 Spectral Clustering

Stuff about Spectral Clustering

2.3.1 The Laplacian Matrix

2.3.2 Normalized Cuts

2.4 Hyperspectral Unmixing

2.4.1 Alternating Direction Method of Multipliers

2.4.2 Global Variable Consensus Optimization using ADMM

2.4.3 Abundance Estimation

2.4.4 Solution Constraints and Regularization Terms

3 Adaptive Superpixel Cuts

3.1 Dataset Preprocessing

3.1.1 Singular Value Decomposition

3.1.2 Layer Normalization

3.2 Algorithm Overview

3.2.1 Superpixel Generation

3.2.2 Construction of the Affinity Matrix W

3.2.3 Normalized Cuts Algorithm

3.2.4 Graph Regularized Fully Constrained Unmixing

3.2.5 Feature Vector Creation

3.2.6 Parameter Selection

4 Experimental Results

4.1 Implementation Details

4.2 Geospatial Testing Datasets

4.2.1 Samson Datasets

4.2.2 Salinas Datasets

4.3 Algorithm Evaluation

4.3.1 Qualitative Evaluation on Samson

4.3.2 Quantitative Evaluation on Salinas

4.4 Algorithm Comparison

5 Applications in Biomedical Imaging

5.1 Dr Yucel Motivations

5.2 Results on Biomedical Hyperspectral Images

6 Conclusions