Report: Segmentation of Natural Images by Texture and Boundary Compression

Daquan Lin 85610653
ShanghaiTech University
Shanghai, China
lindq@shanghaitech.edu.cn

Abstract

This report introduces how to implement the algorithm for segmentation of natural images show in this paper[2]. I test our algorithm on the publicly available Berkeley Segmentation Dataset(BSD)[1].

1. Introduction

In section 2, I will show how to extract the two datasets and some pretreatment. In section 3, I will discuss how to implement K-means model and some problem I faced and how to fix them. Then, I will compare the performances of two datasets. In section 4, I will talk about how to implement simple Mean Shift. Last, come with a conclusion.

2. Texture Encoding

In this operation, I encoded all texture vectors in $\hat{\boldsymbol{X}}$ to represent region R. It seems that I must increase ε to get a good result. So for coding the region R becomes:

$$L_{w,\varepsilon}(R) = \left(\frac{D}{2}+\frac{N}{2w^2}\right)\log_2 det\left(I+\frac{D}{\varepsilon^2}\hat{\sum}_w\right)+\frac{D}{2}\log_2\left(1+\frac{||\hat{\boldsymbol{\mu}}_{\boldsymbol{w}}||^2}{\varepsilon^2}\right) \tag{1}$$

where, $D = 8$, N is the number fo pixels in a region$R$, w is the window size and μ , $\sum$ are the mean and covariance of the verctors in $\hat{\boldsymbol{X}}$.

3. Boundary Encoding

I implemented Freeman chain code by find a start point, and then search next boundary point until find the start point. Next, the coding length $B(R)$is improved by using an adaptive Huffman code that leverages the prior distribution of the chain codes. Suppose an initial orientation (expressed in chain code)o_t, the difference chain code of the following orientation o_{t+1} is $\Delta o_t \doteq mod(o_t - o_{t+1}, 8)$.

$$B(R) = -\sum_{i=0}^{7} \#(\Delta o_t = i)\log_2(P[\Delta o = i]) \tag{2}$$

4. Minimization of the Total Coding Length Function

Suppose an image I can be segmented into non-overlapping regions$\mathcal{R} = \{R_1, \dot{,} R_k\}$, $\cup_{i=1}^k R_i = I$. The total coding length of the image I is

$$L^S_{w,\epsilon}(\mathcal{R}) \doteq \sum_{i=1}^{k} L_{w,\epsilon}(R_i) + \frac{1}{2}B(R_i) \tag{3}$$

The optimal segmentation of I is the one that minimizes$L^S_{w,\epsilon}(\mathcal{R})$. By find the pair of regions R_i and R_j that will maximally decrease if merged:

$$
\begin{aligned}
(R_i, R_j) &= argmax_{R_i, R_j \in \mathcal{R}} \Delta L_{w,\epsilon}(R_i, R_j), where \\
\Delta L_{w,\epsilon}(R_i, R_j) &\doteq L^S_{w,\epsilon}(\mathcal{R}) - L^S_{w,\epsilon}((\mathcal{R} \setminus \{R_i, R_j\}) \cup \{R_i \cup R_j\}) \\
&= L_{w,\epsilon}(R_i) + L_{w,\epsilon}(R_j) - L_{w,\epsilon}(R_i \cup R_j \quad (R_j)
\end{aligned}
\tag{4}
$$

If $\Delta L_{w,\epsilon}(R_i, R_j) > 0$, merge $R_i$and R_j into one region, label as $i = min(i, j)$, and repeat this process, continuing until the coding length $L^S_{w,\epsilon}(\mathcal{R})$ can not be furture reduced.

5. Implementation

Change the RBG image to Lab color space. For each window size $W \in [7, 5, 3, 1]$, apply pixel patch and flat each patch to get the w-neighborhood $W_w(p)$. Define the set of features X by taking the w- neighborhood around each pixel in I, and then stacking the window as a column vector:

$$X = \{\boldsymbol{x}_p \in \mathcal{R}^{3w^2} : \boldsymbol{x}_p = W_w(p)^S for p \in I\} \tag{5}$$

For ease of computation, I further reduce the dimensionality of these features by projecting the set of all features $\boldsymbol{X}$ onto their first $\boldsymbol{D}$ principal components. We denote the set of features with reduced dimensionality as $\hat{\boldsymbol{X}}$. We have observed that for many natural images, the first eight principal components of X contain over 99% of the energy. In this paper, we choose to assign $\boldsymbol{D} = 8$.

For each region or Superpixel in different window size$w$, I generated a map, that is if a region is degenerate$\mathcal{I}_w(R) = \emptyset$, it equal to 1, while equal to 0.

I further construct a region adjacency graph(RAG) by scikit-image package and set each edge weight equal to 1 and repeat merge two regions as shown in Section 4.

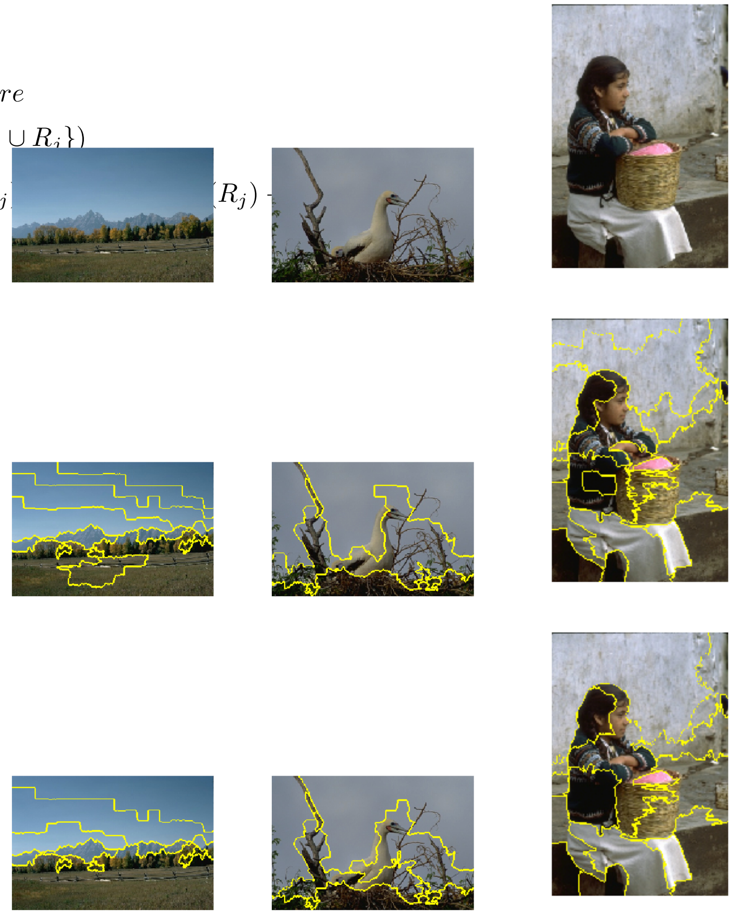

Figure 1: raw 0:$\varepsilon = 0$,raw 1:$\varepsilon = 300$, raw 2:$\varepsilon = 600$

6. Result

7. Conclusion

In this report, I implemented Segmentation of Natural Images by Texture and Boundary Compression. Results show different with different ε.

References

[1] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images

and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proc. 8th Int'l Conf. Computer Vision, volume 2, pages 416–423, July 2001.

[2] H. Mobahi, S. R. Rao, A. Y. Yang, S. S. Sastry, and Y. Ma. Segmentation of natural images by texture and boundary compression. International Journal of Computer Vision, 95(1):86–98, 2011.